ASG-Manager Products Server Facility™ User's Guide

Version 2.5.1

Publication Number: MSF0200-251

Publication Date: November 2001

ASG Worldwide Headquarters Naples Florida USA | asg.com | info@asg.com
1333 Third Avenue South, Naples, Florida 34102 USA Tel: 239.435.2200 Fax: 239.263.3692 Toll Free: 800.932.5536 (USA only)

ASG Documentation/Product Enhancement Fax Form

Please FAX comments regarding ASG products and/or documentation to (239) 263-3692.

Company Name	Telephone Number	Site ID	Contact name

Product Name/Publication	Version #	Publication Date
Product:		
Publication:		
Tape VOLSER:		

Enhancement Request:

ASG Support Numbers

ASG provides support throughout the world to resolve questions or problems regarding installation, operation, or use of our products. We provide all levels of support during normal business hours and emergency support during non-business hours. To expedite response time, please follow these procedures.

Please have this information ready:

- Product name, version number, and release number
- List of any fixes currently applied
- Any alphanumeric error codes or messages written precisely or displayed
- A description of the specific steps that immediately preceded the problem
- The severity code (ASG Support uses an escalated severity system to prioritize service to our clients. The severity codes and their meanings are listed below.)
- Verify whether you received an ASG Service Pack for this product. It may include information to help you resolve questions regarding installation of this ASG product. The Service Pack instructions are in a text file on the distribution media included with the Service Pack.

If You Receive a Voice Mail Message:

1. Follow the instructions to report a production-down or critical problem.
2. Leave a detailed message including your name and phone number. A Support representative will be paged and will return your call as soon as possible.
3. Please have the information described above ready for when you are contacted by the Support representative.

Severity Codes and Expected Support Response Times

Severity	Meaning	Expected Support Response Time
1	Production down, critical situation	Within 30 minutes
2	Major component of product disabled	Within 2 hours
3	Problem with the product, but customer has work-around solution	Within 4 hours
4	"How-to" questions and enhancement requests	Within 4 hours

ASG provides software products that run in a number of third-party vendor environments. Support for all non-ASG products is the responsibility of the respective vendor. In the event a vendor discontinues support for a hardware and/or software product, ASG cannot be held responsible for problems arising from the use of that unsupported version.

Business Hours Support

Your Location	Phone	Fax	E-mail
United States and Canada	800.354.3578	239.263.2883	support@asg.com
Australia	61.2.9460.0411	61.2.9460.0280	support.au@asg.com
England	44.1727.736305	44.1727.812018	support.uk@asg.com
France	33.141.028590	33.141.028589	support.fr@asg.com
Germany	49.89.45716.222	49.89.45716.400	support.de@asg.com
Singapore	65.6332.2922	65.6337.7228	support.sg@asg.com
All other countries:	1.239.435.2200		support@asg.com

Non-Business Hours - Emergency Support

Your Location	Phone	Your Location	Phone
United States and Canada	800.354.3578		
Asia	65.6332.2922	**Japan/Telecom**	0041.800.9932.5536
Australia	0011.800.9932.5536	**Netherlands**	00.800.3354.3578
Denmark	00.800.9932.5536	**New Zealand**	00.800.9932.5536
France	00.800.3354.3578	**Singapore**	001.800.3354.3578
Germany	00.800.3354.3578	**South Korea**	001.800.9932.5536
Hong Kong	001.800.9932.5536	**Sweden/Telia**	009.800.9932.5536
Ireland	00.800.9932.5536	**Switzerland**	00.800.9932.5536
Israel/Bezeq	014.800.9932.5536	**Thailand**	001.800.9932.5536
Japan/IDC	0061.800.9932.5536	**United Kingdom**	00.800.9932.5536
		All other countries	1.239.435.2200

ASG Web Site

Visit http://www.asg.com, ASG's World Wide Web site.

Submit all product and documentation suggestions to ASG's product management team at http://www.asg.com/asp/emailproductsuggestions.asp.

If you do not have access to the web, FAX your suggestions to product management at (239) 263-3692. Please include your name, company, work phone, e-mail ID, and the name of the ASG product you are using. For documentation suggestions include the publication number located on the publication's front cover.

Contents

Preface

This *ASG-Manager Products Server Facility User's Guide* provides information for installing and using the ASG-Manager Products Server Facility module (herein called MPSF). The tasks described in this publication are generally performed by your Systems Administrator. MPSF is an APPC/MVS server that supports access to Manager Products from APPC and TCP/IP clients.

Allen Systems Group, Inc. (ASG) provides professional support to resolve any questions or concerns regarding the installation or use of any ASG product. Telephone technical support is available around the world, 24 hours a day, 7 days a week.

ASG welcomes your comments, as a preferred or prospective customer, on this publication or on any ASG product.

About this Publication

This publication consists of these chapters:

- Chapter 1, "Introduction," provides an overview and the requirements for installation of MPSF.
- Chapter 2, "Using the Data-in-Virtual Access Method," discusses the OS/390 Data-in-Virtual Access Method (DIV).
- Chapter 3, "Getting Started," guides you through the setup and initial logon process.
- Chapter 4, "Using MPSF," provides instructions for monitoring and controlling MPSF for day-to-day use.
- Chapter 5, "The Server Log," discusses the use of the server log and describes the messages sent to the log.
- Chapter 6, "Miscellaneous Server Facilities," describes the optional facilities that can enhance the usability of MPSF.
- Chapter 7, "Server Compatibility," describes the compatibility issues between Manager Products when implemented to execute under MPSF and when implemented for non-MSPF execution.

Related Publications

The documentation library for ASG-Manger Products Server Facility consists of these publications (where *nn* represents the product version number):

- *ASG-DataManager TSO/ISPF Interface Installation* (DMR0300-*nn*-ISPFI) describes the installation of ASG-DataManager's TSO/ISPF Interface facility (selectable unit DMR-FE70). The TSO/ISPF Interface runs ASG-DataManager and ASG-ControlManager under IBM's TSO version of ISPF.
- *ASG-Manager Products Controller's Manual* (MPR2100-*nn*-CON) describes the capabilities and commands provided for the Controller of an ASG-Manager Products dictionary/repository.
- *ASG-Manager Products Installation in OS Environments* (MPR0300-*nn*-INST) this publication is intended for the Systems Administrator, the dictionary Controllers, and those members of the Operations and/or Systems Programming departments who are responsible for installing and running your Manager Products.
- *ASG-Manager Products Message Guide* (MPR2100-*nn*-MES) provides the error messages, explanations, and actions for ASG-Manager Products.
- *ASG-Manager Products Server Facility User's Guide* (MSF0200-*nn*) provides information about the installation and use of the ASG-Manager Products Server Facility module.
- *ASG-Manager Products Systems Administrator's Guide* (MPR2100-*nn*) describes the function and commands of the Systems Administrator at an installation using Manager Products.
- *ASG-ManagerView Installation Instructions* (MVW2600-*nn*) provides software/hardware requirements and the installation procedure for installing and tailoring ASG-ManagerView.
- *ASG-Web Enabler User's Guide* (WBR0200-*nn*) provides information about using ASG-Web Enabler.
- *ASG-Web Enabler Administrator's Guide* (WBR2100-*nn*) provides information for system administrators about installing and configuring ASG-Web Enabler.

Note:

To obtain a specific version of a publication, contact the ASG Service Desk.

JCL Examples

All example JCL and CLISTs shown in this publication are available as members of the dataset MP.JCL.

Publication Conventions

Allen Systems Group, Inc. uses these conventions in technical publications:

Convention	Represents
ALL CAPITALS	Directory, path, file, dataset, member, database, program, command, and parameter names.
Initial Capitals on Each Word	Window, field, field group, check box, button, panel (or screen), option names, and names of keys. A plus sign (+) is inserted for key combinations (e.g., Alt+Tab).
lowercase italic monospace	Information that you provide according to your particular situation. For example, you would replace *filename* with the actual name of the file.
Monospace	Characters you must type exactly as they are shown. Code, JCL, file listings, or command/statement syntax. Also used for denoting brief examples in a paragraph.
Vertical Separator Bar (|) with underline	Options available with the default value underlined (e.g., Y|N).

Syntax Conventions

The following conventions apply to syntax diagrams that appear in this publication.

Diagrams are read from left to right along a continuous line (the "main path"). Keywords and variables appear on, above, or below the main path.

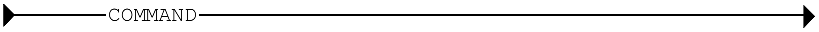
Convention	Represents
▶▶	At the beginning of a line indicates the start of a statement.
▶◀	At the end of a line indicates the end of a statement.
──────▶	At the end of a line indicates that the statement continues on the line below.
▶──────	At the beginning of a line indicates that the statement continues from the line above.

Keywords are in upper-case characters. Keywords and any required punctuation characters or symbols are highlighted. Permitted truncations are not indicated.

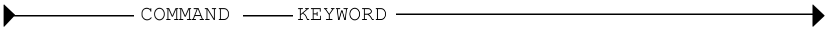
Variables are in lower-case characters.

Statement identifiers appear on the main path of the diagram:

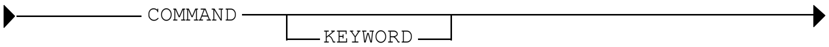
Convention	Represents



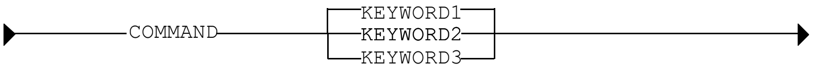
A required keyword appears on the main path:



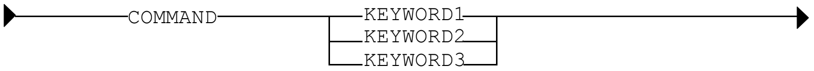
An optional keyword appears below the main path:



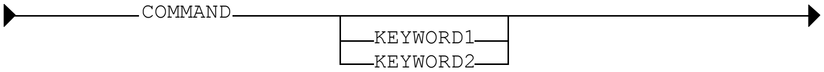
Where there is a choice of required keywords, the keywords appear in a vertical list; one of them is on the main path:



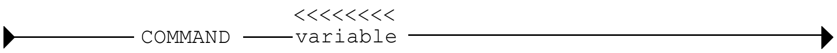
Or



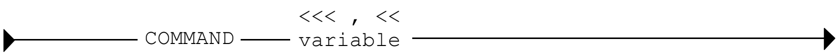
Where there is a choice of optional keywords, the keywords appear in a vertical list, below the main path:



The repeat symbol, <<<<<<, above a keyword or variable, or above a whole clause, indicates that the keyword, variable, or clause may be specified more than once:

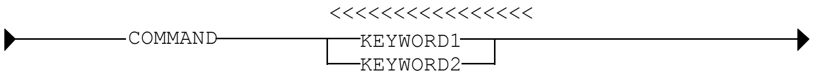


A repeat symbol broken by a comma indicates that if the keyword, variable, or clause is specified more than once, a comma must separate each instance of the keyword, variable, or clause:

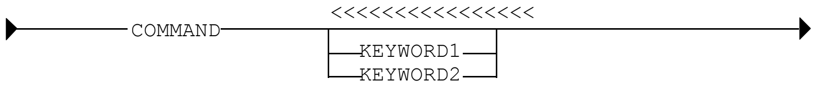


The repeat symbol above a list of keywords (one of which appears on the main path) indicates that any one or more of the keywords may be specified; at least one must be specified:

Convention	Represents



The repeat symbol above a list of keywords (all of which are below the main path) indicates that any one or more of the keywords maybe specified, but they are all optional:



1 Introduction

This chapter introduces MPSF and contains these sections:

MPSF Overview

MPSF is a multi-tasking, multi-user environment that provides a separate and persistent Manager Products environment for each client. It works as an APPC/MVS server supporting access to Manager Products from various client environments including TCP/IP clients.

A typical environment is an OS/390 host running TSO and batch jobs along with access through the internet with a web browser. Multiple servers are permitted, but each must be uniquely named and separately defined to APPC/VTAM.

These are some of the functions performed by MPSF:

- Registers and de-registers as an APPC/MVS server
- Connects to/terminates the TCP/IP interface
- Receives requests for new client conversations
- Creates and starts a Client Processing Task (CPT) for each new conversation
- Terminates client conversations
- Manages timeouts

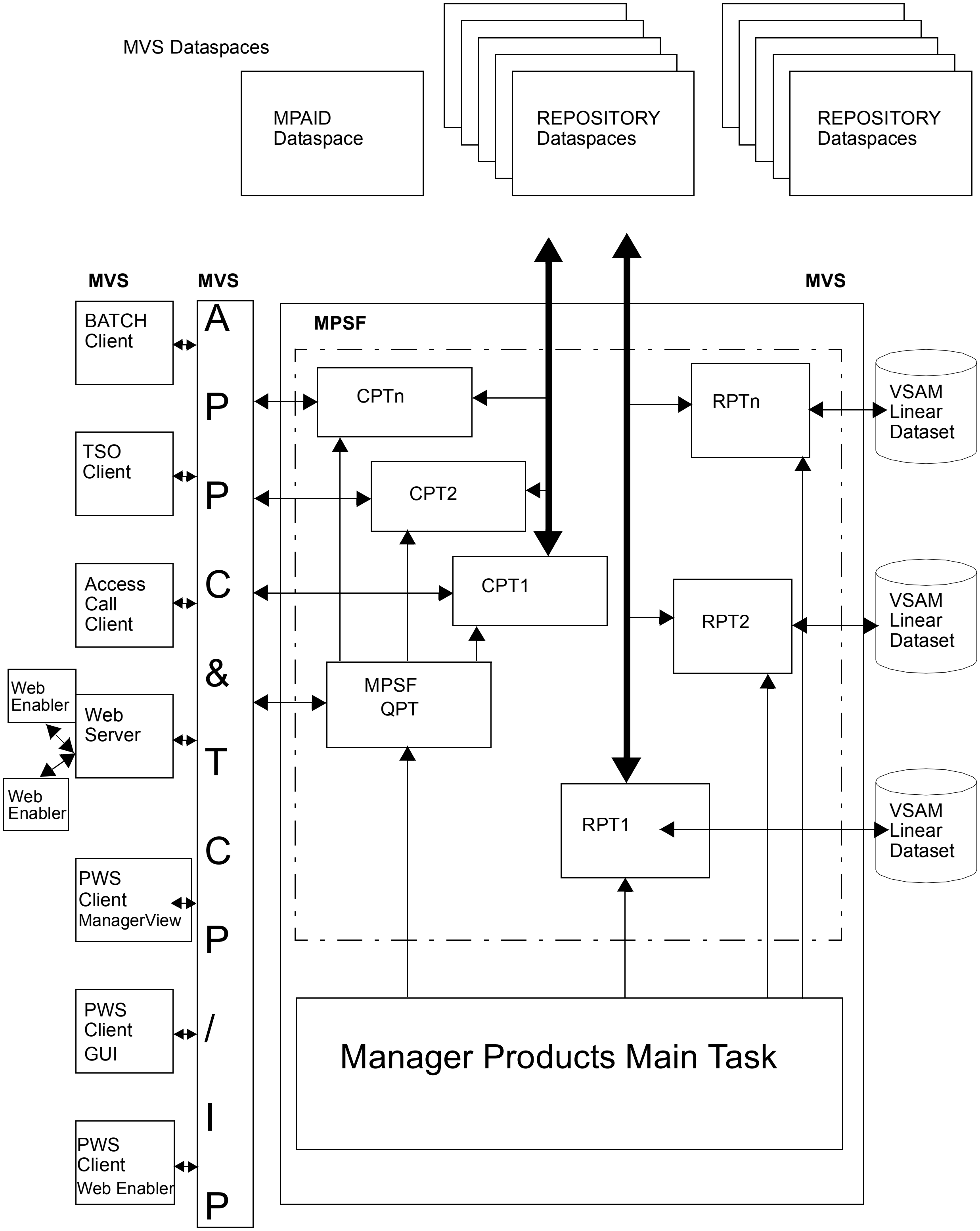
MPSF executes in a standard MVS address space. A request for Manager Product resources results in the creation of an APPC or TCP/IP conversation between the client and MPSF. The Queue Processing Task (QPT) receives and schedules the execution of client requests from APPC. One Client Processing Task (CPT) per active conversation executes client requests.

For multi-tasking, MPSF uses the Systems Administrator subtasking facility with some extensions. You should be familiar with the commands of that facility, since some of them are used to control the execution of Queue and Client Processing Tasks. The server component itself runs under the QPT and is controlled with variants of the SERVER command.

In addition to the scheduling and management of client conversations, MPSF also provides a high speed access method for both repositories and MPAIDs. This access method uses the Data-in-Virtual (DIV) component of OS/390 to update the repository and MPAID data from shared buffers created as OS/390 dataspaces. Each repository and/or MPAID is managed by a Resource Processing Task (RPT) which is responsible for the reading and writing of data from/to a DIV dataset on disk. Using the DIV access method is optional but recommended by ASG for optimum MPSF performance and response times. See Chapter 2, "Using the Data-in-Virtual Access Method," on page 13 for more information.

See Figure 1 on page 3 for the architecture of MPSF.

Figure 1 • MPSF Architecture



Software Requirements

These are the software requirements for installing MPSF:

- MVS or OS/390
- VTAM
- APPC
- TCP/IP is optional, but required for ManagerView and Web Enabler (Servlet Version)
- ASG-MethodManager

  Or

  ASG-ControlManager selectable unit SF01

Virtual Storage Requirements

Virtual storage for all client conversations is allocated from the MPSF address space and increases in direct proportion to the number of concurrent conversations. Virtual storage is mainly allocated above the 16 MB line (31-bit). This section helps you determine a suitable execution class for MPSF. Contact your System Programming Group for more information and assistance in obtaining a suitable address space.

These are the virtual storage requirements for the Manager Products shared program code:

24-bit	300K maximum
31-bit	3MB maximum

These virtual storage values must be added to the requirements given below for each QPT, RPT, and CPT.

Queue Processing Task (QPT)

24-bit Working storage allocated during command processing. 8K maximum is required.

31-bit Allow 1MB for working storage plus any storage allocated by the Manager Diagnostic Facility (MDF). This is an optional feature and if specified has a default buffer allocation of 20MB. Refer to "Manager Diagnostic Facility" on page 84 for further details.

Resource Processing Task (RPT)

24-bit Working storage allocated during command processing. 8K maximum is required.

31-bit Allow 400K for working storage.

The RPT values given here must be multiplied by the number of RPTs required (one for each shared DIV resource).

Client Processing Task (CPT)

24-bit Working storage allocated during command processing. 8K maximum is required. This value increases to 40K if any BDAM repositories or MPAIDs are accessed.

31-bit Working storage allocated during command processing. 500K is sufficient for most commands.

Buffers and directories used with procedures. 500K is a typical value.

The CPT values given here must be multiplied by the expected maximum number of concurrent client conversations. If any non-DIV MPAID or repository access is required, then for *each* client requiring access an allowance must be made for the 31-bit buffer pools needed by these resources. Refer to *ASG-Manager Products Installation in OS Environments* for details of how to calculate these sizes.

Storage shortages adversely effect the usability of MPSF; you may want to add a reasonable margin for safety and increase the amount of virtual storage available to MPSF.

Example

Assume a client requirement of around 1MB with a maximum of 80 concurrent conversations and 5 RPTs. This would be the suggested minimum virtual storage availability:

24-bit	QPT		8K	
	RPT	5 * 8K	40K	
	CPT	80 * 8K	640K	
	Program code		300K	
	Total		988K	= 1MB

31-bit	QPT	1MB + 20MB MDF default buffer	21MB
	RPT	5 * 400K	2MB
	CPT	80 * 1MB	80MB
	Program code		3MB
	Total		106MB

The above details are only a guide to the amount of virtual storage needed. Your needs will vary depending on the commands used and the data accessed.

Maximum Supported Client Conversations

The design of MPSF allows for a maximum of 999 active tasks. This number of active tasks can be achieved if sufficient 24-bit virtual storage is available in an OS/390 address space. Typically, this ranges from 6 to 8MB.

For each MPSF task, whether a QPT, RPT, or CPT, up to 8K of 24-bit storage is currently required. You can therefore calculate an expected maximum number of concurrent client conversations as follows:

```
(24-bit region size in K-300)/8 = maximum number MPSF tasks
```

```
maximum number MPSF tasks-(RPT count+1) = maximum number of client
conversations.
```

Note:

If client access is required to any BDAM repositories or MPAIDs then the storage requirement for those clients is increased to 40K, giving a reduced maximum number of concurrent conversations.

Example

Assume a maximum region size of 8192K (8MB) and an MPSF requirement of 4 RPTs. The expected maximum number of concurrent client conversations is calculated as follows:

```
((8192-300)/8)-(4+1) = 981 maximum client conversations
```

Client Installation and Execution Requirements

MPSF supports two types of clients, those executing in a Programmable Workstation Environment (PWSE) and those executing in a Mainframe Environment (MFE) under the MVS or OS/390 operating systems. MPSF should be up and running before any attempt is made to execute Manager Products on the client side. A number of error codes can be returned when attempting to connect to MPSF. These are documented in the *ASG-Manager Products Message Guide*.

Programmable Workstation Environment (PWSE)

Supported products are:

ASG-Web Enabler. Refer to the *ASG-Web Enabler Systems Administrator's Guide* for full details of installation and execution requirements. If you are using the servlet version of Web Enabler you must enable TCP/IP client support before starting MPSF.

ASG-ManagerView. Refer to the *ASG-ManagerView Installation Instructions* for installation and execution requirements. You must enable TCP/IP client support before starting MPSF.

Mainframe Environment

Supported execution environments are:

- Batch
- TSO
- TSO/ISPF
- Manager Products Access Call
- ROSCOE

Currently excluded environments are CICS and IMS/DC.

No special installation steps are required for the client side execution of Manager Products. It is assumed that the Manager Products loadlib, supplied as MP.LOADLIB, is installed and available. All repositories and MPAIDs to be accessed *must* be allocated by the server start-up job. Additional input or output datasets needed by some Manager Products functionality can be allocated either on the client or server side, but their placement needs to be carefully considered in order to avoid performance degradation or shared usage problems. Refer to "Allocation of External Datasets" on page 10 for further details.

Batch Execution

Use this sample JCL (Figure 2) as the basis for submitting client batch jobs for execution under MPSF:

Figure 2 • Submitting client batch jobs

```
//MPSFBAT  JOB  ,,MSGLEVEL=(1,1),CLASS=A
//MPBATCH  EXEC PGM=MPR00,PARM='SERVER=server'
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
Manager Products commands
.
.
.
//
```

where `server` is the name of the required MPSF server. If a global server name has been defined (see "Tailoring Manager Products to Automatically Execute as an MPSF Client" on page 10), then the PARM field can be omitted, provided that the global server name is correct for your job.

Submitting Batch Jobs Using the WAIT Parameter

You can specify the WAIT parameter in the PARM field of the job EXEC statement to submit batch jobs for execution under MPSF while the server is not currently executing. When the server is not executing, jobs submitted with this parameter will wait for the server to become available instead of terminating with a return code of 16 (unable to connect to server).

You may find this facility useful if, for example, you submit your batch jobs automatically with job scheduling software and you do not want them to fail when the server is temporarily unavailable.

You can use the WAIT parameter alone or in combination with the SERVER= parameter. You can specify parameters in any order.

Parameter examples:

```
PARM=WAIT
PARM='SERVER=MPRSVR00,WAIT'
```

TSO Execution

Use this sample CLIST (Figure 3) as the basis for executing Manager Products in MPSF mode under TSO:

Figure 3 • Executing under TSO

```
/*   Manager Products MPSF CLIST (MPSFTSO)     */
ALLOC FILE(MPIN) DSN(*)
ALLOC FILE(MPOUT)DSN(*)
CALL  'MP.LOADLIB(MPR00)'   'SERVER=server'
FREE  FILE(MPIN,MPOUT)
```

where *`server`* is the name of the required MPSF server. If a global server name has been defined (see "Tailoring Manager Products to Automatically Execute as an MPSF Client" on page 10), then the SERVER= field can be omitted, provided that the global server name is correct for your job.

TSO/ISPF Execution

The TSO/ISPF execution mode is Manager Products executing under the Manager Products selectable unit FE70.

Refer to the *ASG-DataManager TSO/ISPF Interface Installation* for full details of installation and execution requirements.

Access Call Execution

The sample JCL (Figure 2 on page 8) or CLIST (Figure 3 on page 9) used for batch or TSO MPSF execution can be used with a customized EXEC or CALL statement reflecting the name of your Access Call (User Interface) program and omitting the Manager Products parameter specification.

In order for an Access Call program to execute in MPSF mode the DDMR field of the DCONTROL parameter block must reflect the name of the required MPSF server. If a global server name has been defined (see "Tailoring Manager Products to Automatically Execute as an MPSF Client" on page 10), then you do not need to update DDMR as long as the global server name is correct for your job. Specify `NO` or `NOSERVER` in DDMR if you want to bypass MPSF mode.

ROSCOE Execution

Refer to *ASG-Manager Products Installation in OS Environments* for installation and execution requirements.

Tailoring Manager Products to Automatically Execute as an MPSF Client

Executing Manager Products as an MPSF client normally requires a parameter that specifies the server name to use. However, if you want to execute Manager Products mainly using MPSF to a specific server then you can globally specify the server name. Globally specifying the server name avoids having to specify a parameter with every Manager Products execution.

You can use the SERVER keyword of the DCUST macro to specify the name of the required server. Once activated, execution of Manager Products in any of the supported environments attempts to establish a conversation between the client and the specified server. Refer to *ASG-Manager Products Installation in OS Environments* for details of how to specify and generate a revised global options block using the DCUST macro. You can override the global server name specified by providing a parameter with the required name, or you can bypass MPSF execution mode by specifying a server name of `NO`.

An example DCUST specification is:

```
DCUST    SERVER=MPSERV01
```

Allocation of External Datasets

Some Manager Products functionality requires the reading and/or writing of datasets in addition to the repositories and MPAIDs normally used. In MPSF mode all repositories and MPAIDs to be accessed *must* be allocated on the server side. It is *not* possible to access a repository or MPAID allocated on the client side.

You can allocate other datasets on both server and client sides. Better performance is obtained when a dataset is allocated on the server side because data from a client side allocation is always received/sent using APPC communications. However, avoid making server side allocations for output sequential datasets where there is a possibility of multi-user access. If these are made, unpredictable results can occur resulting in a variety of system abends. Server side allocations take priority over identical client side allocations. Use of the IPPF/ISPEXEC command to dynamically allocate a dataset results in a client side allocation. To obtain a server side allocation, use the Systems Administrator’s DSS (Data Set Services) command. Refer to "Dataset Services Facility" on page 88 for further details.

ASG recommends these dataset allocations:

Command/ Facility	Restricted Usage[†]	Input/ Output	Access Method	ASG Recommended Allocation	Alternative Allocation
AUDIT ARCHIVED	Yes	Input	Sequential	Server side	Client side
AUDIT ONTO	Yes	Output	Sequential	Server side	Client side

Command/ Facility	Restricted Usage[†]	Input/ Output	Access Method	ASG Recommended Allocation	Alternative Allocation
CONVERT	No	Input	Partitioned	Server side	Client side
EXTRACT	No	Input	Partitioned	Server side	Client side
EXTRACT	No	Input	Sequential	Server side	Client side
LOG ARCHIVE/ ANALYSIS	Yes	Input/ Output	Sequential	Server side	Client side
POST	No	Output	Sequential	Client side	No
READ	No	Input	Partitioned	Server side	Client side
READ	No	Input	Sequential	Server side	Client side
RECORD/REPLAY	No	Input/ Output	Sequential	Client side	No
RELOAD/LOAD/ RESTORE	Yes	Input	Sequential	Server side	Client side
SAVE/UNLOAD	Yes	Output	Sequential	Server side	Client side
SENDF/ WRITEF	No	Output	Partitioned	Server side	Client side
SENDF/ WRITEF	No	Output	Sequential	Client side	No
Source Language Generation	No	Output	Partitioned	Server side	Client side
Source Language Generation	No	Output	Sequential	Client side	No
SWITCH	No	Output	Sequential	Client side	No
TRACE output	No	Output	Sequential	Client side	No
TRANSFER	No	Output	Partitioned	Server side	Client side
TRANSFER	No	Output	Sequential	Client side	No
WRITE	No	Output	Partitioned	Server side	Client side
WRITE	No	Output	Sequential	Client side	No

[†] **Restricted usage=Yes** indicates a command available only to the Systems Administrator or a repository Controller and where multi-user access is not expected.
Alternative allocation=No indicates that a server side allocation is not recommended.

Using the Attention Key

When using Manager Products under TSO, you can use the terminal attention key to terminate a Manager Products read-only command if necessary.

When executing in MPSF mode, terminal users using the attention key is reflected back to the client side of the Manager Products conversation only. The server side of the conversation detects the request for termination by polling the client side to determine if a command termination has been requested. You can set the polling frequency by issuing this command: `SET ATTENTION-INTERVAL nn;`

where *nn* is an integer in the range 1 to 99 being the requested polling interval, in seconds.

The default polling interval is set to 3 seconds.

Determine the current polling interval by issuing this command: `QUERY ATTENTION-INTERVAL;`.

Implementing Conversation Access Security

By default, when allocating client APPC conversations, no access security information is passed on to the allocate call to APPC.

You may need to pass security information for authorization by the security software in use (e.g. RACF/ACF2) at your installation. MPSF provides a mechanism for you to specify the information to pass.

You can specify a global user ID and password that is passed on all APPC conversation allocations using the MPSFSEC keyword of the Manager Products DCUST macro.

Specify the required user ID and password in this form:

```
DCUST    MPSFSEC=(userid,password)
```

where *userid* and *password* are undelimited strings of up to 8 characters in length.

Refer to *ASG-Manager Products Installation in OS Environments* for details of how to generate an updated global options block (module DMU09) using the DCUST macro.

If you do pass access security information then ensure that SECACPT=CONV is specified in the Manager Products MPSF VTAM APPL definition. See step 3 on page 34 in section "Configuring APPC and VTAM" on page 33 for more information.

2 Using the Data-in-Virtual Access Method

This chapter includes these sections:

Overview

When executing MPSF you can build your repositories and MPAIDs using the Data-in-Virtual (DIV) access method of OS/390. Use of DIV enables high-speed access to Manager Products data by eliminating the processing overheads that can be associated with the use of the standard Manager Products access methods, BDAM and VSAM. DIV repositories and MPAIDs are created on permanent storage as VSAM linear datasets. Creation and population of these DIV objects is normally undertaken as a batch job under the authority of the Systems Administrator or repository Controller. General users can share access and updating by logging on to Manager Products as an MPSF client.

Manager Products data is made available to all MPSF clients from shared buffers created in one or more OS/390 dataspaces. A DIV repository or MPAID can be considered to exist in its entirety as a contiguous stream of data held in virtual storage and available without the need for physical I/O to read the requested data into virtual storage. When data in the buffer is referenced for the first time, a page fault occurs, and the system reads the data from the linear dataset into real storage.

Each DIV repository or MPAID that is to be made available under MPSF must be opened in shared mode from a specified Manager Products subtask. This subtask is considered the owner of that repository or MPAID resource for the duration of MPSF. Such a subtask is a Resource Processing Task (RPT) and is responsible for mapping the linear dataset to the dataspace buffer. The RPT also performs the writing back of updated records from the buffer to the linear dataset. You can start a capture session to write updated records at a user defined interval depending on the frequency and volume of updates to the resource. As with the reading of data, the MPSF client is not involved in performing physical I/O to write updated records back to permanent storage. In both cases, access to the resource is at a logical record level. Eliminating the physical I/O improves the performance and response time and is only limited by the amount of CPU power and real storage that is made available to MPSF.

You do not need to migrate all of your repositories or MPAIDs to use the DIV access method. Under MPSF you can mix existing repositories or MPAIDs created using BDAM or VSAM with those created using DIV, but for maximum performance you should migrate those with any significant usage.

System Requirements

Successful implementation of DIV repositories and MPAIDs requires the usage of certain system resources in addition to those required for non-DIV execution. Check the following requirements with your Systems Programming Group to determine if the required resources are or can be made available for Manager Products use.

Using OS/390 Dataspaces

When creating or opening a DIV repository, Manager Products creates 4 or 5 (5 if logging is specified) OS/390 dataspaces. These dataspaces are used as shared buffers for each of the components of a repository (e.g., Index, Source, Data-entries, Recovery, and, optionally, Log). The size of each dataspace allocated is calculated as the size of each component specified or defaulted at repository creation time plus an overhead of 0.1% of the specified size (minimum overhead is 16K). The overhead is needed by Manager Products for internal control blocks and indexes. The maximum size of any dataspace allocated is 2,000,000K plus an overhead of 2000K. Repository dataspaces allocated are shared by all MPSF clients.

Similarly, when creating or opening a DIV MPAID, Manager Products creates a single OS/390 dataspace to be used as a shared MPAID buffer. The size of the dataspace allocated is calculated as the size of the MPAID specified at MPAID creation time plus an overhead of 0.1% of the specified size (minimum overhead is 16K). This overhead is needed by Manager Products for internal control blocks and indexes. The maximum size of the dataspace allocated is 2,000,000K plus an overhead of 2000K. The MPAID dataspace allocated is shared by all MPSF clients.

Using OS/390 Page Datasets

Manager Products DIV resources can make heavy use of the system page datasets depending on the number and size of DIV resources in concurrent use and the availability of system real storage. Requirements vary according to the number of concurrent MPSF clients, mix of Manager Products commands used, and the demands placed on the system by other non-Manager Product workloads. It may be necessary to add additional page datasets in order to permit very large repositories to be created and subsequently accessed under MPSF.

The requirement for page datasets is likely to be highest during the creation and population of a repository or MPAID. As a minimum requirement there should be sufficient free space within the allocated page datasets to accommodate the total dataspace allocation of the largest repository or MPAID to be created.

Use this formula to calculate the page dataset requirement, in cylinders, when creating a repository or MPAID:

```
(size*1024)/(4096*blkstrk*trkscyl)
```

where:

size is the size in K of the repository or MPAID as specified at creation time and must include the overhead value.

blkstrk is the number of 4096 byte blocks that can be written to a track of the device used for page datasets.

trkscyl is the number of tracks to a cylinder for the device used for page datasets.

For example, assuming a repository size of 720,000K with 3390 devices used for page datasets the space requirement in cylinders is:

```
(720000*1024)/(4096*12*15) = 1000
```

Execution Mode of Manager Products

Manager Products *must* execute from an APF-authorized library in order to successfully make a DIV repository or MPAID available to MPSF clients. Consult your Systems Programming Group to achieve this requirement.

Failure to execute MPSF from an APF-authorized library causes the Resource Processing Tasks (RPTs) of MPSF to terminate with an S047 abend when attempting to open a DIV resource in shared mode.

Creating and Loading of DIV MPAIDs

A DIV MPAID is created by building an MPAID in dataspace virtual storage and writing it to permanent storage as a single VSAM linear dataset. Any type of MPAID, whether for use as a primary read/write MPAID or as a secondary read-only MPAID, can be created. The structure and usage of a DIV MPAID is identical to that of a standard VSAM or BDAM MPAID.

Note:

Although the creator of a DIV MPAID cannot specify a physical blocksize for the dataset (all data in a VSAM linear dataset is stored in 4K blocks) a notational physical blocksize is calculated by Manager Products at create time as the multiple of the specified logical blocksize closest to 32,760 bytes and is used during internal MPAID processing.

If you are creating a read-only concatenated MPAID and want to share it across multiple MPSF instances, you should specify SHAREOPTIONS (2,3) when allocating the VSAM cluster. For more information on sharing MPAIDs, see "Sharing Read-only Concatenated MPAIDs" on page 75.

For further information regarding MPAID content and usage refer to *ASG-Manager Products Systems Administrator’s Guide*.

The loading of an MPAID with your own or ASG-supplied data can be performed in the Manager Products MPAID creation job or in a separate run. As a minimum the ASG-supplied datasets MP.INFO.UNLOAD and MP.COM.UNLOAD should be loaded if not already available on non-DIV MPAIDs or if you wish to create DIV MPAIDs. When using MethodManager you need to make available the content of dataset MP.MMR.UNLOAD. You can load these datasets to one or more DIV MPAIDs, as required. See the *ASG-Manager Products Installation in OS Environments* for further information about these datasets.

A DIV MPAID *must* be allocated with a disposition of old (DISP=OLD) when used in a non-MPSF environment.

To create a DIV MPAID

1 Execute the VSAM service program IDCAMS to allocate the VSAM linear dataset used as permanent storage for the MPAID.

2 Execute Manager Products to create a virtual storage MPAID, which is then written to the VSAM linear dataset. The MPAID VCREATE command must be used to create a DIV MPAID. The syntax for the MPAID VCREATE command is:

```
MPAID  VCREATE ADMINISTRATOR  logon-identifier  PASSWORD  password
       LOGICAL-BLOCKSIZE  nnnn  SIZE  ssss ;
```

where:

logon-identifier is the Systems Administrator logon identifier and is a name (delimited or undelimited) of no more than 10 characters.

password is the Systems Administrator password and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.

nnnn is an integer specifying the required logical blocksize in bytes and must be in the range 600 to 32760.

ssss is an integer specifying the required size (in K) of the MPAID and cannot exceed a value of 2000000.

You can use restricted commands if you are successfully logged on with the password and logon identifier of the Systems Administrator. After creating the MPAID change the password at frequent intervals using the MPAID PASSWORD command for added security.

No default logical blocksize is supplied, but ASG recommends starting with a logical blocksize of around 1000 bytes. Use the MPAID ANALYSE command to help choose an optimum logical blocksize for performance tuning the MPAID.

At creation time the specified MPAID size is rounded up to a multiple of the calculated physical blocksize and may result in an allocation slightly larger than requested.

Use the sample JCL and Manager Products commands (Figure 4 on page 18) as the basis for creating a DIV MPAID:

Figure 4 • Creating a DIV MPAID

```
//MPSF0010 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//ALLOC    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=sysout-class
//SYSIN    DD   *
   DELETE (div-mpaid-dsn) CLUSTER
   DEFINE CLUSTER (NAME(div-mpaid-dsn) -
   KB(kkkk) -
   LINEAR -
   VOLUMES(vvvvvv) -
   SHAREOPTIONS(1,3))
//MPCREATE EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPRDIAG  DD   SYSOUT=sysout-class
//SYSUDUMP DD   SYSOUT=sysout-class
//MPAID    DD   DISP=OLD,DSN=div-mpaid-dsn
MPAID VCREATE ADMINISTRATOR logon-identifier PASSWORD password
LOGICAL-BLOCKSIZE nnnnn SIZE ssss;
LOGON logon-identifier PASSWORD password;
MPAID STATUS;
LOGOFF;
//
```

where:

kkkk is an integer specifying the space in K to be allocated to the VSAM linear dataset. This value must be equal to the size (*ssss*) specified on the MPAID VCREATE command plus a Manager Products overhead of 0.1% of the specified size (minimum 16K). This is needed for internal control blocks and indexes and must be a minimum value of 16K.

vvvvvv is the volume serial number of the disk where the VSAM cluster is to be allocated.

logon-identifier, *password*, *nnnn*, and *ssss* are as previously given for the MPAID VCREATE command.

If you want to load data into the MPAID from an MPAID UNLOAD dataset in the same run, the above job must be modified to include an MPAIDR DD statement for the input unload dataset together with an appropriate MPAID LOAD command placed after the LOGON command. Refer to *ASG-Manager Products Systems Administrator's Guide* for details of the MPAID LOAD command.

Use the sample JCL and Manager Products commands (Figure 5) as the basis for separately loading data to a previously created DIV MPAID:

Figure 5 • Separately loading data to a previously created DIV MPAID

```
//MPSF0020 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//MPLOAD   EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPRDIAG  DD   SYSOUT=sysout-class
//SYSUDUMP DD   SYSOUT=sysout-class
//MPAID    DD   DISP=OLD,DSN=div-mpaid-dsn
//MPAIDR   DD   DISP=OLD,DSN=mpaid-input-unload-dsn
LOGON logon-identifier PASSWORD password;
MPAID LOAD member-type-details;
MPAID STATUS;
LOGOFF;
//
```

where:

member-type-details identifies the MPAID member types to be loaded.

logon-identifier and *password* are as previously given for the MPAID VCREATE command.

Backup and Recovery of DIV MPAID Resources

Creating a Backup of a DIV MPAID

Use the MPAID UNLOAD command to create a backup copy of a DIV MPAID, as you would for a BDAM or VSAM-organized MPAID. See the *ASG-Manager Products Systems Administrator's Guide* for details.

You can create a backup using non-Manager Products software, but ASG does not recommend it.

Recovery of a DIV MPAID

Use the MPAID RELOAD command to recover the content of a DIV MPAID from a Manager Products backup. This is similar to recovery of a BDAM or VSAM-organized MPAID, but there are differences in both the syntax of the command and the job control requirements.

For further general information about the MPAID RELOAD command refer to the *ASG-Manager Products Systems Administrator's Guide*.

The syntax of the MPAID RELOAD command is:

```
MPAID RELOAD PASSWORD password  <SIZE ssss> ;
```

where:

> `password` is the Systems Administrator password and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.
>
> `ssss` is an integer specifying the new size (in K) of the MPAID and cannot exceed a value of 2000000.

The optional SIZE clause can be used to increase or decrease the size of your MPAID. If specified, you must add an additional IDCAMS execution step to the sample JCL shown in Figure 6 to delete and reallocate the existing VSAM cluster to the new size required. This step, together with the specification of the variables, is identical to the IDCAMS execution step shown in Figure 4 on page 18.

Use the sample JCL and Manager Products commands (Figure 6) as the basis for reloading a DIV MPAID where no size change is required:

Figure 6 • Reloading a DIV MPAID

```
//MPSF0025 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//MPRELOAD EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPRDIAG  DD   SYSOUT=sysout-class
//SYSUDUMP DD   SYSOUT=sysout-class
//MPAID    DD   DISP=OLD,DSN=div-mpaid-dsn
MPAID RELOAD PASSWORD password;
LOGON logon-identifier PASSWORD password;
MPAID STATUS;
LOGOFF;
//
```

where:

> `logon-identifier` is the Systems Administrator logon identifier and is a name (delimited or undelimited) of no more than 10 characters.
>
> `password` is the Systems Administrator password and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.

Creating and Populating DIV Repositories

A DIV repository is created by building a repository in dataspace virtual storage and writing it to permanent storage as a single VSAM linear dataset. The structure and usage of a DIV repository is essentially the same as a standard VSAM or BDAM repository, the major difference being that permanent storage for a DIV repository is a single dataset instead of the four or five used for VSAM or BDAM.

Note:

Although the creator of a DIV repository cannot specify a physical blocksize for the Index, Source, or Data-entries components of a repository (all data in a VSAM linear dataset is stored in 4K blocks) a notional physical blocksize is calculated by Manager Products at create time as the multiple of the specified logical blocksize closest to 32760 bytes. This calculation is used during internal repository processing.

For further information about the facilities and commands available to the repository Controller refer to *ASG-Manager Products Controller's Manual*.

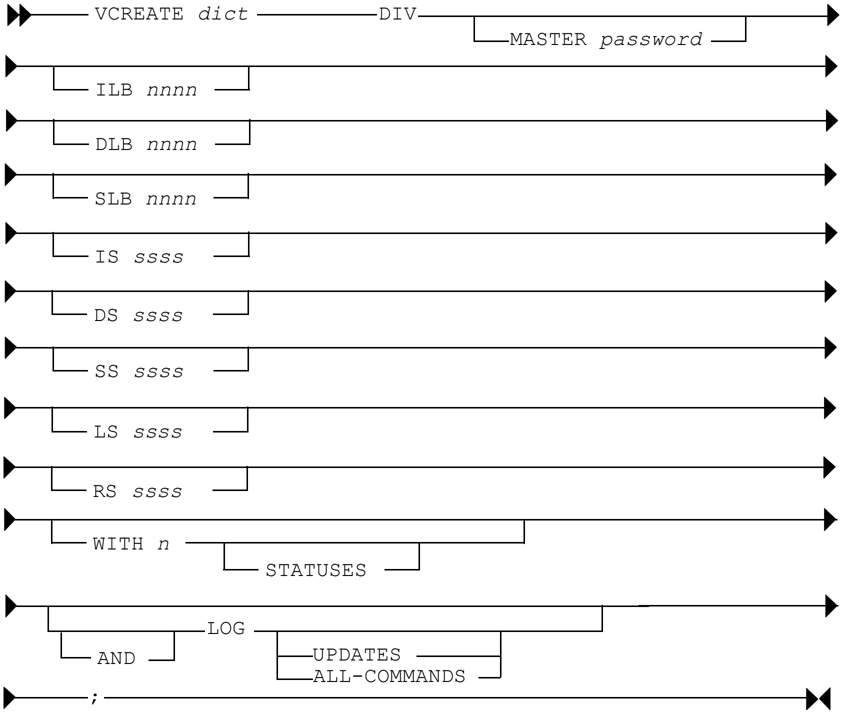
You can populate a DIV repository with your own or ASG-supplied data in the Manager Products repository creation job or in a separate run.

A DIV repository *must* be allocated with a disposition of old (DISP=OLD) when used in a non-MPSF environment. Furthermore, the restricted keyword DIV *must* be specified on the DICTIONARY command used to open the repository in a non-MPSF environment and when opening it for shared usage under MPSF. However, all MPSF clients use the standard form of the command when opening such a repository.

To create a DIV repository

1 Execute the VSAM service program IDCAMS to allocate the VSAM linear dataset used as permanent storage for the repository.

2 Execute Manager Products to create a virtual storage repository which is then written to the VSAM linear dataset. The VCREATE command must be used to create a DIV repository. The syntax for the VCREATE command is:



where:

`dict` is a string of up to 6 printable characters, being the name by which the repository will be known.

`password` is the Controller password for the repository and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.

`nnnn` is an integer specifying a logical blocksize, in bytes.

`ssss` is an integer specifying the required size (in K) of a repository component and cannot exceed a value of 2,000,000.

`n` is an integer in the range 1 to 255.

Default values are provided for all keywords. At the simplest level, a DIV repository can be created using the command: `VCREATE dict DIV;` assuming the defaults are acceptable.

These default values are used:

Keyword	Specifies	Value
IS	Index Space allocation in K	64
SS	Source Space allocation in K	256
DS	Data-entries Space allocation in K	256
RS	Recovery Space allocation in K	64
LS	Log Space allocation in K	64
ILB	Index Logical Blocksize in bytes	2047
SLB	Source Logical Blocksize in bytes	300
DLB	Data-entries Logical Blocksize in bytes	300
WITH	Number of Statuses	1
MASTER	Master Password	dictionary name
LOG	Log requirements	none
none	Recovery Blocksize in bytes	8192
none	Log Blocksize in bytes	8192

The Recovery and Log blocksizes are fixed at 8192 bytes and cannot be modified.

Specifying DIV Component Sizes

DIV component sizes are specified by using these clauses:

```
IS ssss
SS ssss
DS ssss
LS ssss
RS ssss
```

where *ssss* is the required size in K.

Valid abbreviations are used in the above clauses. Following are the valid long forms of each abbreviation:

```
IS is a valid abbreviation of INDEX-SIZE
SS is a valid abbreviation of SOURCE-SIZE
DS is a valid abbreviation of DATA-SIZE
RS is a valid abbreviation of RECOVERY-SIZE
LS is a valid abbreviation of LOG-SIZE
```

At creation time any specified or defaulted sizes are rounded up to a multiple of the calculated physical blocksize and may result in an allocation slightly larger than requested.

The sum of all specified or defaulted sizes cannot exceed 4,000,000 (K).

Specifying Logical Blocksizes

Logical blocksizes are specified using these clauses:

```
ILB nnnn
SLB nnnn
DLB nnnn
```

where *nnnn* is the required logical blocksize in bytes.

Valid abbreviations are used in the above clauses. Following are the valid long forms of each abbreviation:

```
ILB is a valid abbreviation of INDEX-LOGICAL-BLOCKSIZE
SLB is a valid abbreviation of SOURCE-LOGICAL-BLOCKSIZE
DLB is a valid abbreviation of DATA-LOGICAL-BLOCKSIZE
```

The maximum logical blocksize that can be specified is 8192 bytes, while the minimum logical blocksizes are:

INDEX	520 bytes
SOURCE	300 bytes
DATA-ENTRIES	300 bytes

Specifying the Number of Statuses

You can specify the number of statuses that are to be available in a DIV repository, by including this clause in a VCREATE command: `WITH n STATUSES` where *n* is an unsigned integer in the range 1 to 255, specifying the number of statuses required in the repository.

The WITH clause is only accepted if the Basic Status or Advanced Status facility is installed (selectable unit CMR-DD2 or CMR-AD2). If the WITH clause is present in the command but a status facility is not installed, an error message is output and the VCREATE command is not completed.

If the WITH clause is present in the command and a status facility is installed, the DIV repository is created with the specified number of unnamed statuses. These statuses can be named and manipulated subsequently by STATUS commands.

If a status facility is installed, and the WITH clause is omitted from the command, the repository is created with one unnamed status. This status can be named and manipulated subsequently by STATUS commands; but effectively if there is only one status in a repository dictionary the status facility is not used.

Specifying a Log Dataset and Defining What to Log

You can determine whether a log is to be included with a DIV dictionary by including this clause in a VCREATE command: `AND LOG`.

If LOG is present in the command, all subsequent updating commands issued on the repository created by this command, together with their associated member definitions or amendments, are logged.

The optional keyword AND introducing the LOG clause has no processing significance. It is available in the syntax purely for readability of the command.

The optional alternative keywords UPDATES and ALL-COMMANDS which are available with the LOG clause are accepted but ignored if the Audit and Security Facility (selectable unit CMR-DD3) are not included in your Manager Products configuration. If the Audit and Security Facility is installed, use these keywords to override the value of the COMTYPE parameter of the DLOG installation macro. (See *ASG-Manager Products Installation in OS Environments*.)

If LOG is present but neither UPDATES nor ALL-COMMANDS is stated, then the value of the COMTYPE parameter of the DLOG installation macro determines whether the command operates as though LOG UPDATES were stated or as though LOG ALL-COMMANDS were stated.

If LOG UPDATES is stated, all subsequent updating commands issued on the repository created by this command, together with their associated member definitions or amendments, are logged.

If LOG ALL COMMANDS is stated, all subsequent commands issued on the repository created by this command, together with their associated member definitions or amendments, are logged.

The option to log UPDATES or ALL-COMMANDS can be changed after the repository has been created, if required, using a variation of the LOG command.

Use this sample JCL and Manager Products commands (Figure 7) as the basis for creating a DIV repository:

Figure 7 • Creating a DIV repository

```
//MPSF0030 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//ALLOC    EXEC PGM=IDCAMS
//SYSPRINT DD   SYSOUT=sysout-class
//SYSIN    DD   *
   DELETE (div-repository-dsn) CLUSTER
   DEFINE CLUSTER (NAME(div-repository-dsn) -
   KB(kkkk) -
   LINEAR -
   VOLUMES(vvvvvv) -
   SHAREOPTIONS(1,3))
//MPCREATE EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPRDIAG  DD   SYSOUT=sysout-class
//SYSUDUMP DD   SYSOUT=sysout-class
//MPAID    DD   DISP=OLD,DSN=div-mpaid-dsn
//dict     DD   DISP=OLD,DSN=div-repository-dsn
LOGON logon-identifier PASSWORD password;
VCREATE dict DIV MASTER password
ILB nnnn DLB nnnn SLB nnnn
IS ssss DS ssss SS ssss RS ssss LS ssss
WITH n STATUSES
AND LOG;
LOGOFF;
//
```

where:

kkkk is an integer specifying the space in K to be allocated to the VSAM linear dataset. This value must be equal to the sum of the sizes (*ssss*) defaulted or specified on the VCREATE command plus a Manager Products overhead of 0.1% of the total size specified. This is needed for internal control blocks and indexes and must be a minimum value of 16K. The maximum value that should be specified for *kkkk* is 4,000,000 plus the overhead value of 4000.

vvvvvv is the volume serial number of the disk space where the VSAM cluster is to be allocated. For large repositories, you may need to specify more than one volume.

logon-identifier, *password*, *nnnn*, dict, and *ssss* are as previously given.

If you want to populate the repository from a Manager Products SAVE dataset in the same run, then the above job must be modified to include a dictR DD statement for the input SAVE dataset together with an appropriate RESTORE command placed after the VCREATE command. DICTIONARY and AUTHORITY commands are not required as a successful VCREATE command opens the repository automatically. Refer to the *ASG-Manager Products Controller's Manual* for details of the RESTORE command.

Use this sample JCL and Manager Products commands (Figure 8) as the basis for separately restoring data to a previously created DIV repository:

Figure 8 • Separately restoring data to a previously created DIV repository

```
//MPSF0040 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//MPLOAD   EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPRDIAG  DD   SYSOUT=sysout-class
//SYSUDUMP DD   SYSOUT=sysout-class
//MPAID    DD   DISP=OLD,DSN=div-mpaid-dsn
//dict     DD   DISP=OLD,DSN=div-repository-dsn
//dictR    DD   DISP=OLD,DSN=input-save-dsn
LOGON logon-identifier PASSWORD password;
DICTIONARY dict DIV;
AUTHORITY password;
RESTORE restore-selection;
LOGOFF;
//
```

where:

`restore-selection` specifies the data to be restored from the input SAVE dataset

dict, *logon-identifier*, and *password* are as previously given.

Backup and Recovery of DIV Repository Resources

Creating a Backup of a DIV Repository

Use the UNLOAD command to create a backup copy of a DIV repository in the same way as you would for a BDAM or VSAM-organized repository. For further details refer to the *ASG-Manager Products Controller's Manual*.

You can create a backup of a DIV repository created without a log using non-Manager Products software but, ASG does not recommend it.

You *must* use the Manager Products UNLOAD command to create a backup of a DIV repository created with a log. If you do not and it becomes necessary to rebuild the repository, then recovery from non-Manager Products backup overwrites any transactions written to the log since the backup was taken, and roll-forward of the repository to its latest updated state will not be possible.

Recovery of a DIV Repository

Use the RELOAD command to recover the content of a DIV repository from a Manager Products backup. This is similar to recovery of a BDAM or VSAM-organized repository, but there are differences in the syntax and capabilities of the command and the job control requirements.

For further general information about the RELOAD command refer to the *ASG-Manager Products Controller’s Manual*.

You need to consider and make provision for these differences:

- You *must* not delete and reallocate the repository VSAM cluster before execution of the RELOAD command unless you intend to recreate an empty log by using the LOG clause of the RELOAD command. Improper or inadvertent deletion and reallocation of the VSAM cluster inhibits any required roll-forward and complete recovery of the repository to its latest updated state will not be possible. In this situation you will need to create an empty log using the LOG clause of the RELOAD command and manually reapply the deleted transactions.
- You can only use the LOG clause of the RELOAD command if the repository was originally created (using the VCREATE command) with a log.
- The stand-alone command LOG CREATE cannot be used with a DIV repository. If the need to recreate the log component arises, then you must use the RELOAD command with the LOG clause.
- The stand-alone command LOG PURGE cannot be used with a DIV repository. If you want to purge unwanted transactions from the log after an incomplete roll-forward then you must use the RESET-LOG keyword of the RELOAD command. An incomplete roll-forward can occur if the ROLL-FORWARD TO TRANSACTION clause of the RELOAD command is used or if a result code 8 transaction is encountered during roll-forward.

  If you want to purge all transactions on the log after a RELOAD and prevent roll-forward then you can use the NO-ROLL-FORWARD keyword with the RESET-LOG keyword.
- You cannot use the RELOAD command to increase or decrease the size of a DIV repository. If you need to do this, then you must use the SAVE ALL command, followed by VCREATE and RESTORE commands.

Use this command to reload a repository and recover to a particular transaction:

```
RELOAD dict password AND ROLL-FORWARD TO TRANSACTION number RESET-LOG ;
```

All transactions up to and including the specified transaction number are re-applied to the repository. Transactions subsequent to the specified number are purged from the log.

Use this command to reload a repository and drop all transactions from the log:

```
RELOAD dict password NO-ROLL-FORWARD RESET-LOG ;
```

Use this command to reload a repository and roll-forward where there is a result code 8 transaction present:

```
RELOAD dict password RESET-LOG ;
```

All transactions up to but not including the result code 8 transaction are re-applied to the repository. The result code 8 transaction and subsequent transactions are purged from the log.

Use this command to establish a new and empty log component for a reloaded repository:

```
RELOAD dict password AND LOG log-type ;
```

where:

> `dict` is the name of the repository from which the input data set was written.
>
> `password` is the repository Controller or Master Operator password and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.
>
> `number` is an unsigned integer ranging from one to six digits and is the number of a transaction present on the log.
>
> `log-type` specifies the type of logging to be applied and is either UPDATES or ALL-COMMANDS.

Use the sample JCL and Manager Products commands (Figure 9) as the basis for reloading a DIV repository:

Figure 9 • Reloading a DIV repository

```
//MPSF0045 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//MPRELOAD EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPRDIAG  DD   SYSOUT=sysout-class
//SYSUDUMP DD   SYSOUT=sysout-class
//MPAID    DD   DISP=OLD,DSN=div-mpaid-dsn
//dict     DD   DISP=OLD,DSN=div-repository-dsn
//dictR    DD   DISP=OLD,DSN=input-unload-dsn
LOGON logon-identifier PASSWORD password;
RELOAD dict password;
```

```
DICTIONARY dict DIV;
AUTHORITY password;
QUERY DICTIONARY;
LOGOFF;
//
```

where:

`logon-identifier` is the Systems Administrator, repository Controller, or Master Operator Logon Identifier and is a name (delimited or undelimited) of no more than 10 characters.

`password` is the Systems Administrator, repository Controller, or Master Operator password and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.

`dict` is the name of the repository from which the input data set was written.

3 Getting Started

This chapter describes the installation and configuration of MPSF and contains these sections:

Installing Manager Products

Manager Products must be installed and configured as described in *ASG-Manager Products Installation in OS Environments*. If you want to create any non-DIV MPAIDs or repositories then these must be created as described in that publication. The creation of DIV MPAIDs and repositories is described in Chapter 2, "Using the Data-in-Virtual Access Method," on page 13.

Creating a Starter MPAID

An additional Manager Products installation step is required when executing MPSF using a DIV primary MPAID. A DIV MPAID can only be accessed under MPSF via an RPT subtask. You must therefore create a small BDAM (or VSAM) starter MPAID and make the initial logon to Manager Products using this MPAID. Once logged on you can then issue the commands necessary to start the server and make available your DIV resources. The space allocation for the starter MPAID can be minimal as you do not need to load it with any ASG-supplied components, but if you want to start MPSF using Manager Products procedures then you need space in which to store them.

Use this sample JCL and Manager Products commands (Figure 10) as the basis for creating a BDAM MPAID:

Figure 10 • Creating a BDAM MPAID

```
//MPSF0050 JOB  ,,MSGLEVEL=(1,1),CLASS=A
//STARTER  EXEC PGM=MPR00
//STEPLIB  DD   DISP=SHR,DSN=MP.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=sysout-class
//MPAID    DD   DISP=(NEW,CATLG),SPACE=(TRK,1),DSN=starter-mpaid-dsn
MPAID CREATE ADMINISTRATOR logon-identifier PASSWORD password
LOGICAL-BLOCKSIZE 1024 PHYSICAL-BLOCKSIZE 27998;
LOGON logon-identifier PASSWORD password;
MPAID STATUS;
LOGOFF;
/*
```

where:

> `logon-identifier` is the Systems Administrator logon identifier and is a name (delimited or undelimited) of no more than 10 characters.
>
> `password` is the Systems Administrator password and is a string (delimited or undelimited) of no more than 8 printable or non-printable characters.

Refer to *ASG-Manager Products Installation in OS Environments* for details of MPAID creation or for details of VSAM MPAID creation.

Configuring APPC and VTAM

You must update certain APPC and VTAM resource definitions in order for MPSF to function correctly. APPC must allow multiple conversations between Manager Products clients and MPSF registered as an APPC/MVS server. Consult your Systems Programming group to achieve these required changes.

By default client and server code run under the same OS/390 image. The conversations in this mode use a single NOSCHED logical unit (LU) and are LU=OWN conversations.

However, if you want to enable client access from any SNA connected OS/390 image to the MPSF server then you must enable cross-system support as described in step 4 on page 34 after performing step 1 thru step 3 on page 34. The conversations in this mode use the System Base LU as the local LU (client end of conversation) and the LU defined in step 2 and step 3 on page 34 as the partner LU (server end of conversation).

If you are using OS/390 Version 1 Release 3 with VTAM Version 4 Release 4 or later, all APPC/MVS conversations are processed through VTAM; therefore, the Manager Products APPC/MVS LU must be defined to VTAM through an APPL definition statement in SYS1.VTAMLST. For earlier releases the requirements are different.

To configure APPC and VTAM

1 Update the side information dataset.

Add a Manager Products MPSF entry to the APPC/MVS side information dataset for each server you want to execute. These are the required statements:

```
DESTNAME(nnnnnnnn)
TPNAME(MPR00)
PARTNER_LU(nnnnnnnn)
MODENAME(modename)
```

where:

nnnnnnnn is a string of up to eight characters representing the server to be started. It must match the name specified on the SERVER START command used to begin execution of MPSF for this definition (see "Activating MPSF" on page 44).

modename specifies the logon mode for the SNA session connecting the local LU with the partner LU. Use any logon mode valid for your site.

2 Define the LU to APPC/MVS.

In the appropriate APPCPM*xx* member of SYS1.PARMLIB, define a local NOSCHED LU using an LUADD statement. This is the required statement:

```
LUADD ACBNAME(nnnnnnnn)
      NOSCHED
```

where *nnnnnnnn* is a string of up to eight characters representing the server to be started. This is the name that *must* be specified on the SERVER START command used to begin execution of MPSF for this definition (see "Activating MPSF" on page 44) and must be the same value of *nnnnnnnn* specified in step 1.

3 Define the LU to VTAM.

Define the Manager Products MPSF LU to VTAM by adding an APPL definition to SYS1.VTAMLST. This is the required statement:

```
nnnnnnnn APPL ACBNAME=nnnnnnnn,APPC=YES
```

where *nnnnnnnn* is a string of up to eight characters representing the server to be started. This name *must* be specified on the SERVER START command used to begin execution of MPSF for this definition (see "Activating MPSF" on page 44) and must be the same value of *nnnnnnnn* specified in step 1 on page 33 and step 2 on page 33.

You may need to add other parameters to this definition to conform to your site's standard and to ensure that the required number of concurrent sessions is available (keyword DSESLIM).

4 Enable MPSF cross-system support using the XSYSSVR keyword of the Manager Products DCUST macro. You must specify XSYSSVR=YES. Refer to the *ASG-Manager Products Installation in OS Environments* for details of how to generate an updated global options block (module DMU09) using the DCUST macro.

For each OS/390 image from which client access is required, you must ensure that:

A System Base LU is available for use as the conversation local LU. The base LU is defined to APPC/MVS as in step 2 on page 33. The required statement is:

```
    LUADD   ACBNAME(basename)   BASE
```

The base LU must also be defined to VTAM as in step 3. The required statement is:

```
    basename   APPL   ACBNAME=basename,APPC=YES
```

where *basename* is any valid name that conforms to your site's standards.

You may need to add other parameters to this statement to conform to your site's standards and to ensure that the required number of concurrent sessions is available (keyword DSESLIM).

The side information entries for MPSF as defined in step 1 on page 33 are available to APPC/MVS. No further action is required if you can share the dataset in which they are defined across your OS/390 images. If not, then you must define the required entries in an existing or new side information dataset and make it available to APPC/MVS.

You can use the OS/390 system command `DISPLAY APPC,LU,ALL` to determine the LU types and names available to APPC/MVS for the OS/390 image under which the command was issued. The output from this command also gives the name of the active side information dataset, if any.

Starting the Server

Overview and Execution Requirements

You must complete the installation of Manager Products and the configuration of APPC/VTAM before starting the server. These procedures are discussed in "Installing Manager Products" on page 31 and "Configuring APPC and VTAM" on page 33. The repositories and MPAIDs you wish to access must also be available. If DIV repositories or MPAIDs are to be used then your Manager Products program library must be APF-authorized as discussed in "Execution Mode of Manager Products" on page 16.

To successfully start MPSF

1. Log on to the starter MPAID.

2. Create the Manager Products subtasking environment.

3. Start the required RPT subtasks and make shared DIV resources available. These tasks own and manage your DIV resources. An RPT is required for each DIV object.

4. Start the QPT subtask.

5. Activate MPSF using the SERVER command.

You should execute MPSF as a batch job or started task. Ensure that MPSF execution will not be terminated because CPU or line limits are exceeded. You must also ensure that an appropriate System Resource Management (SRM) profile is used to obtain the desired MPSF performance and response times.

You need to provide as a minimum JCL DD statements for MPIN and MPOUT and a DD statement for your starter MPAID. You should not use a DDNAME of MPAID for your starter MPAID as MPSF clients will use the default DDNAME of MPAID when logging on. Select a suitable DDNAME and use it with the MPAID-DDNAME clause of the LOGON command when making the initial logon.

You must also provide JCL DD statements for:

- All DIV MPAIDs and repositories that MPSF tasks will access using DISP=OLD or DISP=SHR. Only users running as MPSF clients can access your DIV datasets. DIV datasets cannot be allocated in any other OS/390 address space while allocated under MPSF except for secondary MPAIDs that will be accessed as read-only concatenations. You can share read-only concatenated MPAIDs across multiple MSPF instances. See "Sharing Read-only Concatenated MPAIDs" on page 75 for details.
- Any non-DIV (BDAM or VSAM) repositories or MPAIDs you want to access. Allocate these using DISP=SHR to allow access by other non-MPSF tasks, if required.
- Other external datasets to be accessed by MPSF clients and can be allocated with any desired disposition. As a general rule, you should not allocate any output sequential datasets where access may be required by more than one MPSF client. Refer to "Allocation of External Datasets" on page 10 for further information on client/server dataset allocation.

You must supply a parm field of SERVER=NO if Manager Products has been configured via the DCUST macro to automatically execute as an MPSF client.

Appendix A,"Example Batch Start" on page 93 is an example of the JCL and Manager Products commands for executing MPSF in batch.

Logging on to the Starter MPAID

Log on to your starter MPAID using a standard LOGON command with an MPAID-DDNAME clause specified. The DDNAME specified on this clause must be the same as the DDNAME used on the JCL DD statement specifying the starter MPAID. For example:

```
LOGON STARTUP PASSWORD STARTUP MPAID-DDNAME STARTER;
```

Creating the Subtasks

Manager Products subtasking is fully explained in *ASG-Manager Products Systems Administrator's Guide*. For working with MPSF, you need to define enough subtasks to process the maximum number of concurrent clients you expect on the server, plus one or more RPT subtasks and one subtask for the QPT. The maximum allowable number of subtasks is 999.

You should not define more subtasks than can be supported in your environment. Refer to "Virtual Storage Requirements" on page 4, which describes how to determine the maximum number of concurrent client conversations that can be supported in your MPSF address space.

To create the server subtasking environment, use this command:

```
SUBTASK DEFINE nnn SERVER;
```

where *nnn* is any number between 2 and 999. Message 10499I is issued when the environment has been successfully created.

Starting the RPT Subtasks

You must start an RPT for each DIV repository or MPAID to be made available under MPSF. Each RPT manages a SINGLE DIV resource, either a repository or MPAID. When a resource is opened in shared mode, then that resource is available to other RPTs, and, after the server is started, to MPSF clients. The order in which resources are opened and made available is therefore important when starting your RPTs. Secondary read/only MPAIDs must be opened first, followed by the primary read/write MPAID and ending with your repositories. When terminating MPSF, your RPTs must be shutdown in reverse order; that is, repositories first, followed by primary read/write MPAID and ending with secondary MPAIDs.

Do *not* start an RPT for any non-DIV (BDAM or VSAM) repository or MPAID.

You can start a maximum of 30 RPTs for any MPSF instance. If you exceed this limit, message 14609E is issued and sharing of the DIV resource fails.

Starting a Secondary MPAID RPT

Start a secondary MPAID RPT by sending commands to the subtask in which you wish the RPT to execute. Any available subtask can be used. To start the RPT, use these commands:

```
SUBTASK START nnn
'LOGON logon-id PASSWORD password MPAID-DDNAME ddname SHARE';
SUBTASK WAIT  nnn;
```

where:

nnn is the required subtask number or name.

logon-id is the logon identifier of the Manager Products Systems Administrator for this secondary MPAID.

password is the logon password of the Manager Products Systems Administrator for this secondary MPAID.

ddname is the DDNAME used on the JCL DD statement defining this MPAID.

The SHARE keyword is mandatory here and causes the DIV resource to be added to the MPSF table of shared resources.

The SUBTASK WAIT command is used to ensure completion of the previously issued command.

By default, a secondary MPAID is opened in read-only mode. Under MPSF, a DIV MPAID is assumed to be a secondary MPAID when the LOGON command sent to the RPT contains an MPAID-DDNAME clause and the DDNAME is not specified as MPAID.

If you want to open the MPAID in update mode to use as a primary MPAID by MPSF clients then you must specify the UPDATE-MPAID keyword of the LOGON command.

When opened read-only a secondary MPAID can only be accessed by MPSF clients in read-only mode.

The procedure described here must be repeated for each secondary MPAID to be made available.

Starting the Primary MPAID RPT

Start the primary MPAID RPT by sending commands to the subtask in which you wish the RPT to execute. Any available subtask can be used. To start the RPT, use these commands:

```
SUBTASK START nnn 'LOGON logon-id PASSWORD password SHARE';
SUBTASK WAIT  nnn;
SUBTASK START nnn ‘MPAID CONCATENATION ddname, ddname, . . .';
SUBTASK WAIT  nnn;
SUBTASK START nnn 'CAPTURE START ttt';
SUBTASK WAIT  nnn;
```

where:

nnn is the required subtask number or name.

logon-id is the logon identifier of the Manager Products Systems Administrator for this primary MPAID.

password is the logon password of the Manager Products Systems Administrator for this primary MPAID.

ddname is the DDNAME used when previously starting one or more secondary MPAID RPTs.

ttt is a CAPTURE session parameter value.

The SHARE keyword is mandatory here. The SHARE keyword causes the DIV resource to be added to the MPSF table of shared resources and the executing subtask to be identified as an RPT.

The SUBTASK WAIT command is used to ensure completion of the previously issued commands.

The meaning and usage of a CAPTURE session is fully explained in "DIV CAPTURE Sessions" on page 40.

Execution of the MPAID CONCATENATION command given here is not mandatory but, in installations where the concatenated MPAID facility is employed, will be required in order to make available the ASG-supplied components needed for some functionality.

Starting a Repository RPT

Start a repository RPT by sending commands to the subtask in which you wish the RPT to execute. Any available subtask can be used. To start the RPT, use these commands:

```
SUBTASK START nnn 'LOGON logon-id PASSWORD pass1 RPT' ;
SUBTASK WAIT  nnn;
SUBTASK START nnn 'MPAID CONCATENATION ddname, ddname, . . .';
SUBTASK WAIT  nnn;
SUBTASK START nnn 'DICTIONARY dictname DIV SHARE';
SUBTASK WAIT  nnn;
SUBTASK START nnn 'AUTHORITY pass2';
SUBTASK WAIT  nnn;
SUBTASK START nnn 'CAPTURE START ttt';
SUBTASK WAIT  nnn;
```

where:

nnn is the required subtask number or name.

logon-id is the logon identifier of the Manager Products Systems Administrator for your primary MPAID.

pass1 is the logon password of the Manager Products Systems Administrator for your primary MPAID.

ddname is the DDNAME used when previously starting one or more secondary MPAID RPTs.

dictname is the name of the repository to be opened.

pass2 is the master (controller) password for the repository to be opened.

ttt is a CAPTURE session parameter value.

You can use the optional keyword RPT of the LOGON command to identify the subtask as an RPT and ensure that all output generated is retained in the RPT output buffer. Identification of the subtask as a repository RPT is normally triggered by receipt of the SHARE keyword of the DICTIONARY command.

The restricted keyword DIV, available only to the Systems Administrator or Guest Controller, must be used to successfully open a DIV repository. MPSF clients, when accessing the repository, use the standard form of the DICTIONARY command.

The SHARE keyword is mandatory here and causes the DIV resource to be added to the MPSF table of shared resources and executing subtask to be identified as an RPT.

The READ keyword of the DICTIONARY command can be used to open the repository in read-only mode. When opened read-only the repository can only be accessed by MPSF clients in read-only mode.

The SUBTASK WAIT command is used to ensure completion of the previously issued commands.

The meaning and usage of a CAPTURE session is fully explained in "DIV CAPTURE Sessions" on page 40.

Execution of the MPAID CONCATENATION command given here is not mandatory but, in installations where the concatenated MPAID facility is employed, will be required in order to make available the ASG-supplied components needed for some functionality.

The procedure described here must be repeated for each repository to be made available.

DIV CAPTURE Sessions

When MPSF clients update DIV repositories and MPAIDs, the updates made are written to the shared buffers created in OS/390 dataspaces. Such changes are immediately reflected to all other MPSF clients and management tasks (RPTs).

For performance reasons updates made by MPSF clients are not written back to permanent storage on disk (a VSAM Linear dataset). Instead, the RPT performs this function on behalf of the clients. Once the DIV resource is opened, you can start an RPT CAPTURE session, which detects client updates and writes updated records to the DIV object on disk. You can specify the frequency at which the CAPTURE session detects updates. CAPTURE sessions are not mandatory, as all updates are written back to disk when the DIV resource is closed, but for all active repositories and primary MPAIDs, ASG *strongly* recommends that you start one.

Caution! With no CAPTURE session active, a hardware or software failure could lead to the loss of a large number of updates.

A CAPTURE session can be ended using a CAPTURE STOP command and the update detection frequency can be amended at any time using CAPTURE STOP/CAPTURE START commands.

You must end an active CAPTURE session before attempting to close the DIV resource.

A CAPTURE session can not be started for a DIV resource opened in read-only mode.

Use the QUERY CAPTURE command to display all active CAPTURE sessions. See "The QUERY CAPTURE Command" on page 62 for further information.

CAPTURE Session Exit

You can add your own processing to a CAPTURE session. An exit is provided, in the form of a Corporate Executive, which allows you to execute Manager Products commands if you wish. The exit is called immediately after the CAPTURE session detects and writes back any updates, whether or not updates are found. Functions are provided which return DIV resource type, name, and update count. You could, for example, implement an automatic repository or MPAID backup process using the UNLOAD command, triggered after a certain number of updates have occurred.

MPCMCAPXIT is the name of the ASG-supplied dummy Corporate Executive and is supplied in the release dataset MP.COM.UNLOAD. The repository source definition is also supplied in the release dataset MP.CORP. Your processing requirements must be added to the repository definition and CONSTRUCTed to an appropriate MPAID.

For the first call of a CAPTURE session to the exit a parameter of START is passed and for the last call a parameter of END is passed.

Procedures Language functions particularly applicable to a user written exit are:

DIVCAPT (args)	Returns CAPTURE session information.
DIVOBJN( )	Returns DIV object name.
DIVOBJT( )	Returns DIV object type.
EXTRACT(MPUPDC)	Returns primary MPAID current update count.
EXTRACT(REPUPDC)	Returns repository current update count.

See *ASG-Manager Products Procedures Language* for complete information.

Starting a CAPTURE Session

Start a CAPTURE session by sending the CAPTURE command to the subtask in which the RPT is executing. To start the session, use this command:

```
SUBTASK START nnn 'CAPTURE START ttt';
```

where:

nnn is the required subtask number or name.

ttt is the required interval (in seconds) between update detection attempts and must be an integer in the range 1 to 3600.

If no frequency is specified then a default value of 60 seconds is used.

Ending or Modifying a CAPTURE Session

A CAPTURE session is normally ended during the termination phase of MPSF. See "Terminating the RPT Subtasks" on page 70 for further details.

Termination is also necessary during execution of MPSF if you want to vary the update detection interval. You need to end the current CAPTURE session and start another with the new interval. See "Communicating with a Server RPT" on page 50 for further details.

The syntax of the command is:

```
CAPTURE STOP;
```

The Wait for Capture Command

MPSF clients do not normally need to be concerned with the physical updating of the DIV resource on permanent storage. This function is automatically performed by the appropriate RPT on behalf of all MPSF clients.

However, there may be situations during the execution of certain critical Manager Products procedures where it is essential to ensure that updates to one or more DIV resources have actually been written back to permanent storage before further execution of the procedure can take place.

Using the Wait for Capture (WFC) command achieves this requirement. This command, available only within a Manager Products procedure, suspends execution of a procedure until all currently updated records in the dataspaces belonging to the specified DIV resource have been physically written back to permanent storage by the CAPTURE session of the appropriate RPT.

The syntax of the WFC command is:

```
    WFC <resource-name> ;
```

where *resource-name* is the name of the DIV repository or MPAID for which update capture is requested and can be up to 8 characters in length. The parameter is optional and defaults to the currently open repository.

These are the return codes issued:

Return Code	Description
0	Updates have been written to permanent storage.
8	An error occurred during WFC processing. An appropriate error message is output that describes the specific error.

Starting the QPT Subtask

Start the QPT by sending commands to the subtask in which you wish the QPT to execute. Any available subtask can be used. To start the QPT, use these commands:

```
SUBTASK START nnn 'LOGON logon-id PASSWORD password QPT';
SUBTASK WAIT  nnn;
SUBTASK START nnn 'MPAID CONCATENATION ddname, ddname, . . .';
SUBTASK WAIT  nnn;
```

where:

nnn is the required subtask number or name.

logon-id is the logon identifier of the Manager Products Systems Administrator for your primary MPAID.

password is the logon password of the Manager Products Systems Administrator for your primary MPAID.

ddname is the DDNAME used when previously starting one or more secondary MPAID RPTs.

You can use the optional keyword QPT of the LOGON command to identify the subtask as the MPSF QPT and ensure that all output generated is retained in the QPT output buffer. Identification of the subtask as the MPSF QPT is normally triggered by receipt of the SERVER command.

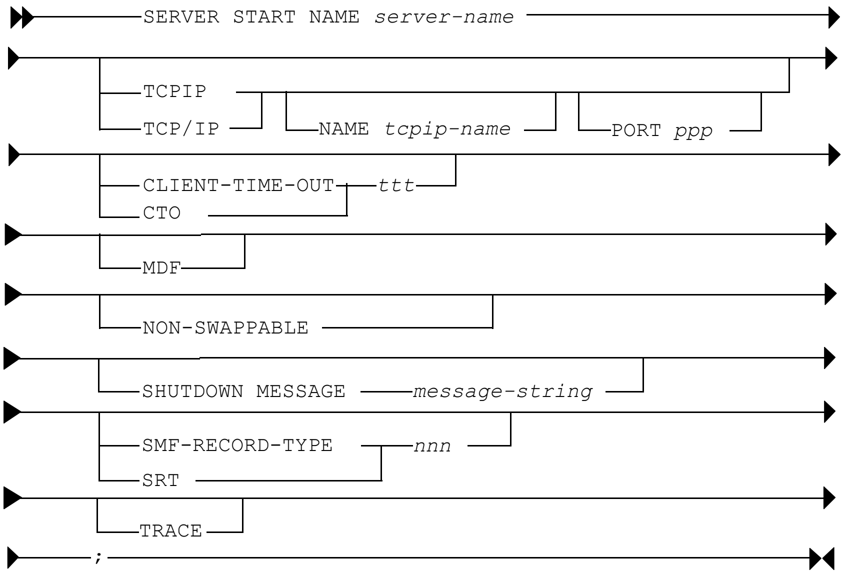
You *must* log on as the Systems Administrator in order to execute the restricted SERVER command needed to activate MPSF (see "Activating MPSF" on page 44).

Execution of the MPAID CONCATENATION command given here is not mandatory but, in installations where the concatenated MPAID facility is employed, will be required in order to make available the ASG-supplied components needed by QPT processing.

Repository access in not needed for successful execution of the QPT. Savings in virtual storage use will also be made if a repository is not opened.

Activating MPSF

To activate MPSF, send a SERVER START command to the QPT. The syntax for the SERVER START command is:



where:

`server-name` is a string of up to eight characters representing the server to be started. It must match the name specified in the first step of installation (see step 1 on page 35).

`tcpip-name` is the name of your TCP/IP address space.

`ppp` is the TCP/IP port number to be used by MPSF and must be in the range 256 to 32000.

`ttt` is an integer in the range 1 to 1440 representing the global time-out value in minutes.

`nnn` is an integer in the range of 120 to 255 representing an SMF record type.

`message-string` is a delimited string of up to 70 characters.

Enable support for TCP/IP conversations to use Manager Products PWS-based clients with MPSF.

If a TCP/IP address space name is not specified, then TCP/IP is assumed.

If a TCP/IP port number is not specified, 2500 is assumed.

If the optional CLIENT-TIME-OUT keyword is omitted then a default value of 30 minutes is used. For further information refer to "Setting a Timeout Limit" on page 64.

When specified the optional MDF keyword enables the Manager Diagnostic Facility. If enabled, any Manager Products diagnostic dumps produced during the execution of MPSF are written to virtual storage where they can be viewed and manipulated by the Systems Administrator. For further information refer to "Manager Diagnostic Facility" on page 84.

To make the MPSF address space non-swappable you can use the optional NON-SWAPPABLE keyword. For further information, refer to "Making the Server Non-swappable" on page 67.

Use the optional SHUTDOWN-MESSAGE clause to display the specified message string to full screen TSO clients when a non-immediate server termination is requested. For further information, refer to "MPSF Shutdown" on page 49.

The optional SMF-RECORD-TYPE clause enables the generation of MPSF client SMF records. For further information, refer to "Writing System Management Facilities (SMF) Records" on page 81.

The optional TRACE keyword is intended for use in problem determination situations only, in conjunction with the ASG Service Desk. When enabled, a client/server communications trace is written to a wrap-around virtual storage buffer. Do not use this keyword for day-to-day execution as performance of MPSF may be adversely effected.

For example, if the QPT is to be started in subtask 1, the name of the server is MPSERV01, TCP/IP support is not required but MDF is to be enabled then you should issue this command:

```
SUBTASK START 1 'SERVER START NAME MPSERV01 MDF';
```

The QPT subtask performs these actions when it receives the SERVER START command:

- Ensures that no other MPSF of any name is active in any Manager Products subtask belonging to the current address space.
- Ensures that no other MPSF with the same name is active in any address space of the current OS/390 image.
- Allocates the virtual storage areas required for the MPSF environment.
- Registers MPSF to APPC/MVS as an APPC server.
- Starts the interface with your TCP/IP address space, if TCP/IP support is specified.
- Enables the QPT component.

Messages 14500I and 14525I are sent to the QPT log when QPT is successfully activated. If TCP/IP support is specified, messages 14526I and 14527I are also sent to the log when TCP/IP support is successfully started. Message 14528I is sent to the QPT log when the Operator Interface is successfully enabled. QPT then waits for conversation allocation requests from APPC, and optionally, from TCP/IP. At this point, MPSF is enabled to service requests from Manager Products clients.

In order to improve readability and ease of use, you can supply the delimited SERVER START command in multiple delimited strings, up to a maximum of 255 characters. Ensure that the appropriate spaces between keywords and values are supplied as necessary.

For example:

```
SERVER START NAME MPRS001 MDF;

SERVER START NAME MPSVR20 TCPIP NAME TCP001 PORT 10000 CTO 90;
```

4 Using MPSF

This chapter describes the commands used and output available for day-to-day use of MPSF. The procedure for starting MPSF is given in "Starting the Server" on page 35. The sections in this chapter include:

Communicating with the Server QPT

You can control MPSF by sending the appropriate SUBTASK commands to the QPT subtask. Do this by logging on as an MPSF client and issuing the commands given below. Any output generated by these commands is appended to the MPSF log (see Chapter 5, "The Server Log," on page 77 for more information). Commands issued to the QPT are also shown in the QPT log. Typically, you communicate with the QPT when you want to vary an MPSF execution parameter, such as the global timeout value. Other Manager Products commands can also be issued, if required.

To send a command to the QPT, use the SUBTASK QSTART command:

```
SUBTASK QSTART nn 'manager-products-command';
```

To view the MPSF log, use the SUBTASK QVIEW command:

```
SUBTASK QVIEW nn;
```

where *nn* is the number or name of the QPT subtask.

For example, to set the global timeout value to 40 seconds with the QPT executing in subtask 1, issue this command:

```
SUBTASK QSTART 1 'SERVER CLIENT-TIME-OUT 40';
```

You can check the results by viewing the MPSF log, using this command:

```
SUBTASK QVIEW 1;
```

The Operator Interface

MPSF's Operator Interface facility allows your Operations personnel to monitor server activity and initiate MPSF shutdown via a system console. Commands issued by an operator are routed to the server QPT for interpretation and execution.

MPSF Monitoring

Note:

For the command syntax given in this section *job-name* is the name of the MPSF batch job or started task. F and P can be used as single character forms of the MODIFY and STOP commands respectively.

To obtain a list of all clients currently logged on to the server enter:

```
MODIFY  job-name,Q
```

The output from this command is the same as that output by the QUERY SERVER command issued by an MPSF client. See "The QUERY SERVER Command" on page 51 for an explanation of the output from this command.

To obtain a list of all current MPSF tasks enter:

```
MODIFY  job-name,T
```

The output from this command is the same as that output by the SUBTASK LIST command issued by an MPSF client. See "The SUBTASK LIST Command" on page 55 for an explanation of the output from this command.

Invalid syntax entered gives this response:

```
INVALID MPSF PARAMETER HAS BEEN IGNORED
```

MPSF Shutdown

Note:

For the command syntax given in this section *job-name* is the name of the MPSF batch job or started task. F and P can be used as single character forms of the MODIFY and STOP commands respectively.

To initiate an immediate shutdown of MPSF enter:

```
MODIFY   job-name,I
```

Or

```
STOP   job-name
```

A successfully issued immediate request to shutdown the server outputs this response:

```
MPSF IMMEDIATE TERMINATION IN PROGRESS
```

To initiate a non-immediate shutdown enter:

```
    MODIFY   job-name,E
```

A successful non-immediate shutdown request can be followed by a later immediate request, if needed.

A successfully issued non-immediate request to shutdown the server outputs this response:

```
    MPSF TERMINATION IN PROGRESS
```

Invalid syntax entered gives this response

```
    INVALID MPSF PARAMETER HAS BEEN IGNORED
```

Communicating with a Server RPT

You can control an RPT subtask during MPSF execution by sending the appropriate SUBTASK commands to the required RPT subtask. You do this by logging on as an MPSF client and issuing the commands given below. Any output generated by these commands is appended to an RPT output buffer, maintained for each RPT. Commands issued to an RPT are also shown in the RPT output buffer. Typically, you communicate with an RPT in order to vary a CAPTURE session interval but other Manager Products commands can also be issued.

To send a command to an RPT, use the SUBTASK RSTART command:

```
SUBTASK RSTART nn 'manager-products-command';
```

To view the RPT buffer, use the SUBTASK RVIEW command:

```
SUBTASK RVIEW nn;
```

where *nn* is the number or name of the RPT subtask.

For an RPT with an active CAPTURE session, pending commands are not processed until after the next capture attempt is made, so there is normally a delay between the issuing and processing of commands sent to an RPT. Use the SUBTASK RVIEW command to ensure that a previously sent command has completed before sending another.

For example, to set a CAPTURE session interval to 100 seconds for an RPT executing in subtask 4, issue the following commands:

```
SUBTASK RSTART 4 'CAPTURE STOP';
SUBTASK RSTART 4 'CAPTURE START 100';
```

You can check the results by viewing the generated output, using this command:

```
SUBTASK RVIEW 4;
```

Monitoring the Server

You should monitor the MPSF status and workload from time to time to detect and correct any error conditions or unexpected events. Failure to take remedial action in a timely manner results in the deterioration of server response time and availability.

The commands QUERY SERVER and SUBTASK LIST help you monitor MPSF. They can be issued from any Manager Products environment connected to MPSF.

The QUERY SERVER Command

The QUERY SERVER command provides you with general and statistical information about MPSF activity and information about each current conversation. This is the command syntax:

```
QUERY SERVER;
```

Figure 11 • Example output from the QUERY SERVER command

```
MANAGER PRODUCTS SERVER MPSERV01 ACTIVITY ON 20 MAY 1999 AT 11.38.25
------------------------------------------------------------------------------
CLIENT    CLIENT  SUBTASK   CONVERSATION  DATA   DATA    MSGS   MSGS
IDENTITY  MODE    IDENTITY  START TIME    IN(K)  OUT(K)  IN     OUT
------------------------------------------------------------------------------
DEVMXC    TSO      TASK-03   11.22.09          1     85     75     27
DEVMXC11  BAT      TASK-04   11.37.10          1     21      4      4
WEBOUT    WE-C     TASK-05   11.35.08          1    101      8      8
DEVMXC07  BAT      TASK-06   11.35.08          1    195      8     18
WEBOUT    WE-C     TASK-07   11.36.07          1     61      6      6
DEVMXC08  BAT      TASK-08   11.36.08          1     81      6      9
DEVMXC13  BAT      TASK-09   11.37.12          1     41      4      8
WEBOUT    WE-C     TASK-10   11.47.13          1     41      4      8
------------------------------------------------------------------------------
SERVER NAME                        : MPSERV01
SERVER START DATE/TIME             : 20 MAY 1999/07.49.01
SERVER EXECUTION STATUS            : ACTIVE
SERVER QUEUE PROCESSING SUBTASK  : TASK-01
CURRENT CONVERSATIONS              :     8
HIGHEST CONCURRENT CONVERSATIONS :     8
MAXIMUM CONCURRENT CONVERSATIONS :    79
TOTAL APPC ALLOCATE REQUESTS       :    60
ACCEPTED APPC ALLOCATE REQUESTS  :    60
REJECTED APPC ALLOCATE REQUESTS  :     0
TOTAL TCP/IP CONNECT REQUESTS      :    40
ACCEPTED TCP/IP CONNECT REQUESTS :    40
REJECTED TCP/IP CONNECT REQUESTS :     0
FREE SUBTASKS AVAILABLE            :    70
GLOBAL CLIENT TIME-OUT (MINUTES) :    60
TOTAL MESSAGES INPUT               :   820
TOTAL MESSAGES OUTPUT              :  1003
TOTAL DATA INPUT (K)               :    58
TOTAL DATA OUTPUT (K)              :  7581
ON-LINE USERS SHUTDOWN MESSAGE     :  PLEASE LOGOFF NOW, MAINTENANCE REQUIRED
------------------------------------------------------------------------------
```

This is the general information provided by the QUERY SERVER command:

Field	Description
Server Name	The name of this MPSF (as specified by the SERVER START command).
Server Start Date/Time	The date and time when MPSF execution started.
Server Execution Status	The current status of MPSF (ACTIVE, SUSPENDED, or in SHUTDOWN mode).
QPT Subtask Name	The name of the subtask where the QPT is executing.

Field	Description
Current Conversations	The number of conversations currently executing.
Highest Concurrent Conversations	The highest number of concurrent conversations since MPSF started.
Maximum Concurrent Conversations	The maximum number of concurrent conversations permitted. This value is calculated from the number of available subtasks unless set by the SERVER MAX-CONV command.
Total APPC Allocate Requests	The total number of APPC conversation allocation requests received since MPSF started.
Accepted APPC Allocate Requests	The number of accepted APPC conversation allocation requests received since MPSF started.
Rejected APPC Allocate Requests	The number of rejected APPC conversation allocation requests received since MPSF started.
Total TCP/IP Connect Requests	The total number of TCP/IP connect requests received since MPSF started.
Accepted TCP/IP Connect Requests	The number of accepted TCP/IP connect requests received since MPSF started.
Rejected TCP/IP Connect Requests	The number of rejected TCP/IP connect requests received since MPSF started.
Free Subtasks Available	The number of subtasks available for use by QPT to service new allocation requests.
Global Timeout Value	The global value (in minutes) to be used for client timeout when no local value has been set with the SET CLIENT-TIMEOUT command.
Total Messages Input	The cumulative total messages input for this MPSF session (not including current conversations).
Total Messages Output	The cumulative total messages output for this MPSF session (not including current conversations).
Total Data Input	The cumulative total data input (in K) for this MPSF session (not including current conversations).

Field	Description
Total Data Output	The cumulative total data output (in K) for this MPSF session (not including current conversations).
Shutdown Message	The text of the shutdown message sent to online users when server shutdown is requested.

This is the information provided about each current conversation:

Field	Description
Client Identity	Identity of the client holding a conversation with Manager Products. Can be the batch job name, TSO logon ID, etc.
Client Mode	The execution mode of the client. These are the possible values: ACC Access Call program BAT Batch job ISPF TSO/ISPF (SU FE70) MVW ManagerView PWS PWS based client REXX REXX Interface ROS ROSCOE TSO TSO full screen WE-C Web Enabler (CGI Version) WE-S Web Enabler (Servlet Version)
Subtask Identity	The name of the subtask where the Client Processing Task (CPT) is executing.
Conversation Start	The time the conversation was allocated.
Data In	The volume of data received from the client (in K).
Data Out	The volume of data sent to the client (in K).
Messages In	A count of the messages received from the client.
Messages Out	A count of the messages sent to the client.

The SUBTASK LIST Command

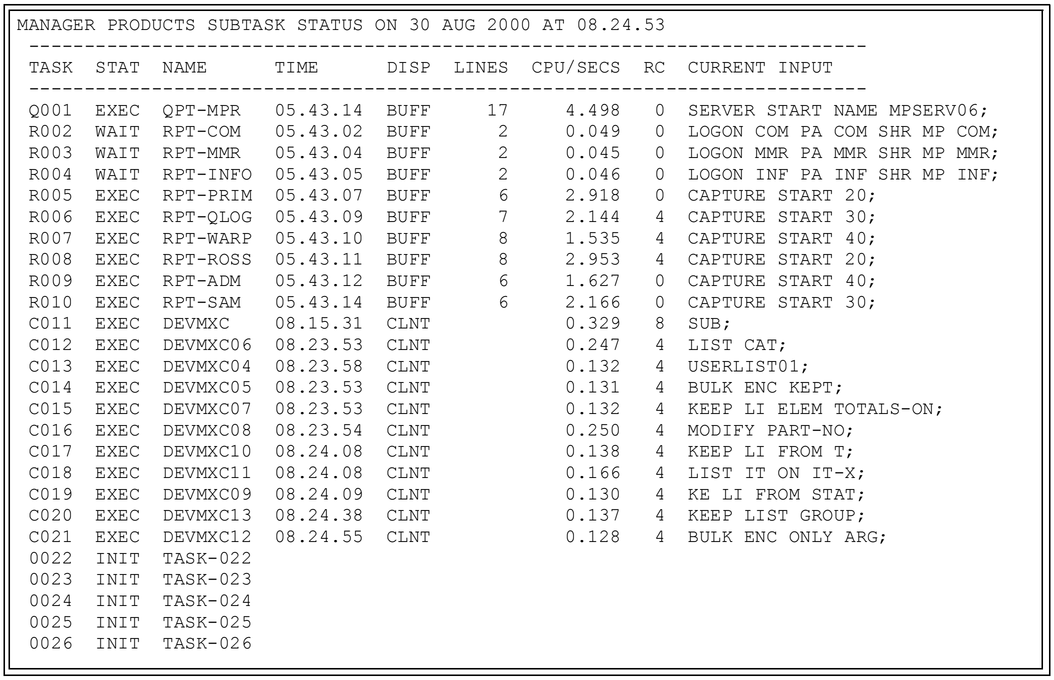
Use the SUBTASK LIST command to examine the CPU usage of a particular CPT, or to determine the currently executing Manager Products command. The output generated by this command is described fully in the *ASG-Manager Products Systems Administrator's Manual*. This is the command:

```
SUBTASK LIST server;
```

where `server` is an optional keyword needed only when the command is issued from an environment where a subtasking environment for the user has been established. If the keyword is omitted, then output from the command relates to the user subtasking environment and not to the server subtasking environment.

If a subtask is executing as a CPT, then CLNT is output in the DISP field of the subtask detail line. No output line count is given for these subtasks. More useful data volumes and message counts are given as output by the QUERY SERVER command (see "The QUERY SERVER Command" on page 51). The task name field is also replaced with the Client Identity. For any QPT, RPT, or CPT the task position of the task number field is replaced with the character Q, R, or C respectively.

Figure 12 • Example output from the SUBSTASK LIST command



```
MANAGER PRODUCTS SUBTASK STATUS ON 30 AUG 2000 AT 08.24.53
 -------------------------------------------------------------------------
 TASK  STAT  NAME      TIME      DISP  LINES  CPU/SECS  RC  CURRENT INPUT
 -------------------------------------------------------------------------
 Q001  EXEC  QPT-MPR   05.43.14  BUFF     17     4.498   0  SERVER START NAME MPSERV06;
 R002  WAIT  RPT-COM   05.43.02  BUFF      2     0.049   0  LOGON COM PA COM SHR MP COM;
 R003  WAIT  RPT-MMR   05.43.04  BUFF      2     0.045   0  LOGON MMR PA MMR SHR MP MMR;
 R004  WAIT  RPT-INFO  05.43.05  BUFF      2     0.046   0  LOGON INF PA INF SHR MP INF;
 R005  EXEC  RPT-PRIM  05.43.07  BUFF      6     2.918   0  CAPTURE START 20;
 R006  EXEC  RPT-QLOG  05.43.09  BUFF      7     2.144   4  CAPTURE START 30;
 R007  EXEC  RPT-WARP  05.43.10  BUFF      8     1.535   4  CAPTURE START 40;
 R008  EXEC  RPT-ROSS  05.43.11  BUFF      8     2.953   4  CAPTURE START 20;
 R009  EXEC  RPT-ADM   05.43.12  BUFF      6     1.627   0  CAPTURE START 40;
 R010  EXEC  RPT-SAM   05.43.14  BUFF      6     2.166   0  CAPTURE START 30;
 C011  EXEC  DEVMXC    08.15.31  CLNT            0.329   8  SUB;
 C012  EXEC  DEVMXC06  08.23.53  CLNT            0.247   4  LIST CAT;
 C013  EXEC  DEVMXC04  08.23.58  CLNT            0.132   4  USERLIST01;
 C014  EXEC  DEVMXC05  08.23.53  CLNT            0.131   4  BULK ENC KEPT;
 C015  EXEC  DEVMXC07  08.23.53  CLNT            0.132   4  KEEP LI ELEM TOTALS-ON;
 C016  EXEC  DEVMXC08  08.23.54  CLNT            0.250   4  MODIFY PART-NO;
 C017  EXEC  DEVMXC10  08.24.08  CLNT            0.138   4  KEEP LI FROM T;
 C018  EXEC  DEVMXC11  08.24.08  CLNT            0.166   4  LIST IT ON IT-X;
 C019  EXEC  DEVMXC09  08.24.09  CLNT            0.130   4  KE LI FROM STAT;
 C020  EXEC  DEVMXC13  08.24.38  CLNT            0.137   4  KEEP LIST GROUP;
 C021  EXEC  DEVMXC12  08.24.55  CLNT            0.128   4  BULK ENC ONLY ARG;
 0022  INIT  TASK-022
 0023  INIT  TASK-023
 0024  INIT  TASK-024
 0025  INIT  TASK-025
 0026  INIT  TASK-026
```

Monitoring Virtual Storage Usage and Availability

Manager Products is highly dynamic in its use of virtual storage. The requirements can vary enormously depending on the products used, repository size, and commands executed. These are just a few of the many factors that influence virtual storage use. Requests to the operating system to obtain virtual storage and subsequently to release it (using OS/390 GETMAIN/FREEMAIN requests) are relatively expensive and in a highly active MPSF environment can account for a significant percentage of the resources consumed by MPSF. You can greatly improve performance and response times by minimizing the number of storage request SVCs executed by MPSF clients.

Manager Products itself tries to reduce the number of storage requests made to the operating system by retaining a pool of free virtual storage to satisfy subsequent user requests. A pool is acquired for each MPSF client, the maximum size is controlled using the Systems Administrator SET FREEPOOL command. By default, the maximum size of the free pool is set to 128K. Refer to *ASG-Manager Products Systems Administrator's Guide* for details of this command.

The QUERY SERVER STORAGE command provides you with statistical information about MPSF virtual storage usage by current tasks. General information is also provided about virtual storage availability, real storage usage, and total dataspace virtual storage allocated to Data-in-Virtual (DIV) resources.

You should use the information provided to ensure that MPSF has sufficient virtual storage available to satisfy the requirements of existing and new clients. ASG recommends restarting MPSF with a larger region size if the total virtual storage available consistently drops below 50MB.

In addition, you need to monitor the ratio between virtual storage requests resolved by operating system SVC and those resolved from the task free pool. You should aim to achieve 80% of requests from task free pools. If this percentage is not achieved then consider increasing the maximum free pool size for a client or group of clients where their current free pool size is at or close to their maximum free pool size. Likely candidates for an increase in the maximum free pool size are TSO clients, in particular those executing MethodManager.

This is the general information provided by the QUERY SERVER STORAGE command:

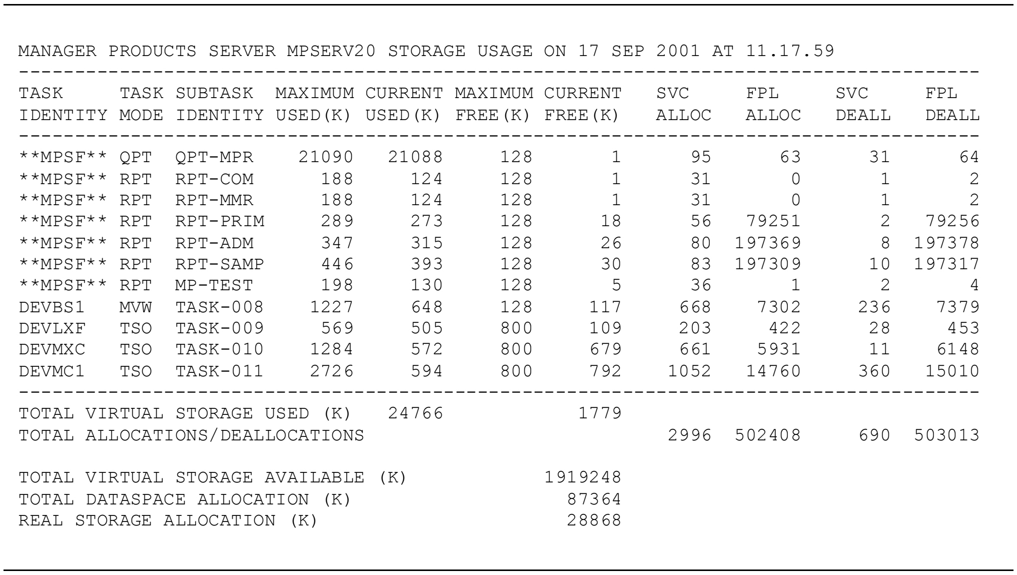
Field	Description
Total Virtual Storage Used	Total virtual storage in use/in free pools by current MPSF tasks, in K.
Total Virtual Storage Allocations	Total virtual storage allocations made by current MPSF tasks. Separate counts are provided for SVC allocations and those made from task free pools.
Total Virtual Storage Deallocations	Total virtual storage deallocations made by current MPSF tasks. Separate counts are provided for SVC deallocations and those where the storage was returned to the task free pool.
Total Virtual Storage Available	Total virtual storage available in MPSF address space, in K.
Total Dataspace Allocation	Total dataspace virtual storage allocated to currently open DIV resources (repositories and MPAIDs), in K.
Total Real Storage In Use	Total real storage allocated to the MPSF address space, in K.

This information is provided for each MPSF task:

Field	Description
Task Identity	MPSF server/client task identity. **MPSF** is shown for an MPSF QPT or RPT server management task.
Task Mode	Execution mode of MPSF server/client task. Shown as QPT or RPT for server management tasks. For MPSF clients, values are as Client Mode of QUERY SERVER command output.
Subtask Identity	MPSF subtask name if specified by SUBTASK NAME command for server management tasks. For MPSF clients, shown as TSK-*nnn*.
Maximum Virtual Storage Used	Maximum virtual storage, in K, used by task.
Current Virtual Storage Used	Current virtual storage, in K, used by task.
Maximum Free Pool Size	Maximum size, in K, of task free pool, as defaulted or specified by the SET FREEPOOL command.
Current Free Pool Size	Current size, in K, of task free pool.
Allocations Using SVC	Count of virtual storage requests made using operating system services.

Field	Description
Allocations From Free Pool	Count of virtual storage requests resolved from task free pool.
Deallocations Using SVC	Count of virtual storage deallocations made using operating system services.
Deallocations To Free Pool	Count of virtual storage deallocations where the storage was returned to the task free pool.

Figure 13 • Output from the QUERY SERVER STORAGE command



```
MANAGER PRODUCTS SERVER MPSERV20 STORAGE USAGE ON 17 SEP 2001 AT 11.17.59
-------------------------------------------------------------------------------------
TASK     TASK SUBTASK  MAXIMUM CURRENT MAXIMUM CURRENT   SVC     FPL     SVC     FPL
IDENTITY MODE IDENTITY USED(K) USED(K) FREE(K) FREE(K)   ALLOC   ALLOC   DEALL   DEALL
-------------------------------------------------------------------------------------
**MPSF** QPT  QPT-MPR    21090   21088     128       1      95      63      31      64
**MPSF** RPT  RPT-COM      188     124     128       1      31       0       1       2
**MPSF** RPT  RPT-MMR      188     124     128       1      31       0       1       2
**MPSF** RPT  RPT-PRIM     289     273     128      18      56   79251       2   79256
**MPSF** RPT  RPT-ADM      347     315     128      26      80  197369       8  197378
**MPSF** RPT  RPT-SAMP     446     393     128      30      83  197309      10  197317
**MPSF** RPT  MP-TEST      198     130     128       5      36       1       2       4
DEVBS1   MVW  TASK-008    1227     648     128     117     668    7302     236    7379
DEVLXF   TSO  TASK-009     569     505     800     109     203     422      28     453
DEVMXC   TSO  TASK-010    1284     572     800     679     661    5931      11    6148
DEVMC1   TSO  TASK-011    2726     594     800     792    1052   14760     360   15010
-------------------------------------------------------------------------------------
TOTAL VIRTUAL STORAGE USED (K)   24766            1779
TOTAL ALLOCATIONS/DEALLOCATIONS                           2996  502408     690  503013

TOTAL VIRTUAL STORAGE AVAILABLE (K)            1919248
TOTAL DATASPACE ALLOCATION (K)                   87364
REAL STORAGE ALLOCATION (K)                      28868
```

Monitoring DIV Resources

Commands are provided which enable you to monitor the availability and usage of your DIV resources. At any point in time you can see which resources are available under MPSF and determine the MPSF clients accessing them.

You can also obtain statistics about active CAPTURE sessions, useful in setting an optimum CAPTURE session update detection interval.

The commands QUERY SHARED and QUERY CAPTURE help you monitor your DIV resources. They can be issued from any Manager Products environment connected to MPSF.

The QUERY SHARED Command

The QUERY SHARED command provides you with information about each shared DIV resource available under the current execution of MPSF together with information about each MPSF client or QPT/RPT subtask currently accessing it. The syntax for the basic command is: `QUERY SHARED;`

For each shared resource the following output is produced:

For the resource owning RPT a single detail line is output giving:

Resource name Name of the DIV resource.

Resource type Either MPD for an MPAID or REP for a repository.

User count Count of MPSF clients accessing the resource.

User name Owning RPT subtask name.

Logon-id ControlManager logon identifier.

Access Either RW (update) or RO (read-only). If the RPT has any control of the resource then RW or RO will be suffixed with:

- S (shared control)
- SL (shared control LUW)
- E (exclusive control)
- EL (exclusive control LUW)

Access date/time Date and time the resource was shared.

If there are any current users for the shared resource then for each user a single detail line is output giving:

User name For a QPT or RPT this is the subtask name and for a CPT is the client identity.

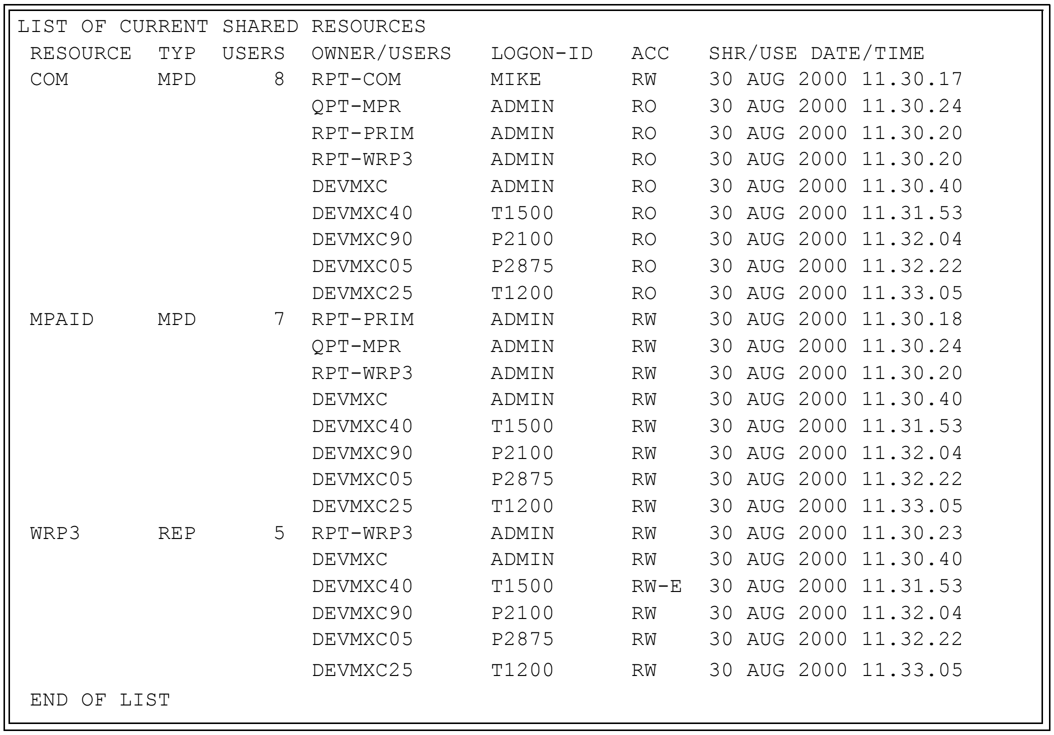
Logon-id ControlManager logon identifier.

Access Either RW (update) or RO (read-only). If user has any control of resource then RW or RO will be suffixed with:

- S (shared control)
- SL (shared control LUW)
- E (exclusive control)
- EL (exclusive control LUW)

Access date/time Date and time of initial access to resource.

Figure 14 • Example output from the QUERY SHARED command



```
LIST OF CURRENT SHARED RESOURCES
 RESOURCE  TYP  USERS  OWNER/USERS   LOGON-ID   ACC    SHR/USE DATE/TIME
 COM       MPD      8  RPT-COM       MIKE       RW     30 AUG 2000 11.30.17
                       QPT-MPR       ADMIN      RO     30 AUG 2000 11.30.24
                       RPT-PRIM      ADMIN      RO     30 AUG 2000 11.30.20
                       RPT-WRP3      ADMIN      RO     30 AUG 2000 11.30.20
                       DEVMXC        ADMIN      RO     30 AUG 2000 11.30.40
                       DEVMXC40      T1500      RO     30 AUG 2000 11.31.53
                       DEVMXC90      P2100      RO     30 AUG 2000 11.32.04
                       DEVMXC05      P2875      RO     30 AUG 2000 11.32.22
                       DEVMXC25      T1200      RO     30 AUG 2000 11.33.05
 MPAID     MPD      7  RPT-PRIM      ADMIN      RW     30 AUG 2000 11.30.18
                       QPT-MPR       ADMIN      RW     30 AUG 2000 11.30.24
                       RPT-WRP3      ADMIN      RW     30 AUG 2000 11.30.20
                       DEVMXC        ADMIN      RW     30 AUG 2000 11.30.40
                       DEVMXC40      T1500      RW     30 AUG 2000 11.31.53
                       DEVMXC90      P2100      RW     30 AUG 2000 11.32.04
                       DEVMXC05      P2875      RW     30 AUG 2000 11.32.22
                       DEVMXC25      T1200      RW     30 AUG 2000 11.33.05
 WRP3      REP      5  RPT-WRP3      ADMIN      RW     30 AUG 2000 11.30.23
                       DEVMXC        ADMIN      RW     30 AUG 2000 11.30.40
                       DEVMXC40      T1500      RW-E   30 AUG 2000 11.31.53
                       DEVMXC90      P2100      RW     30 AUG 2000 11.32.04
                       DEVMXC05      P2875      RW     30 AUG 2000 11.32.22
                       DEVMXC25      T1200      RW     30 AUG 2000 11.33.05
 END OF LIST
```

An extended form of the QUERY SHARED command provides information about usage of the DIV resource in terms of logical accesses to the DIV dataspace buffer for each MPSF client, RPT, or QPT. The syntax of the extended command is:

```
QUERY SHARED USAGE <resource-name>;
```

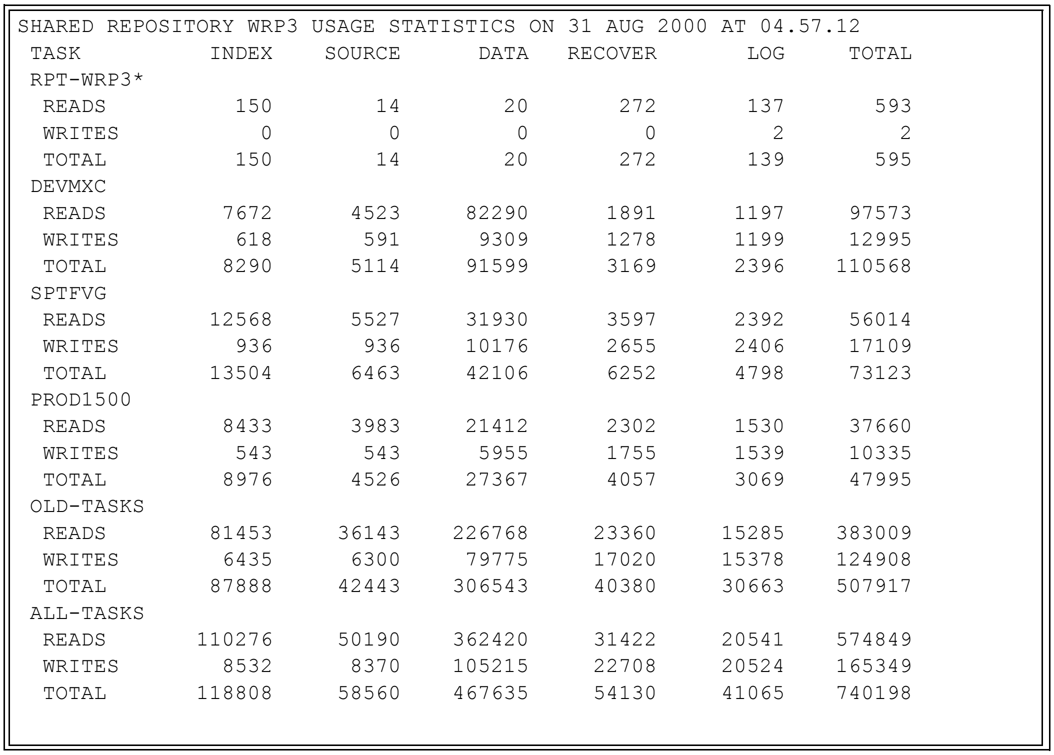
where *resource-name* represents the name of a currently shared DIV resource and is used to limit the output of the command to the specified resource name.

An additional user OLD-TASKS is always shown in the output from this command and represents the total logical accesses performed by completed tasks since the resource was made available for shared usage.

An asterisk (*) to the right of the task name indicates the owning RPT for the resource.

Usage of a resource by a subtask of an MPSF client is indicated by *nn* to the right of the client identity where *nn* is the client subtask number.

Figure 15 • Example output from the QUERY SHARED USAGE command

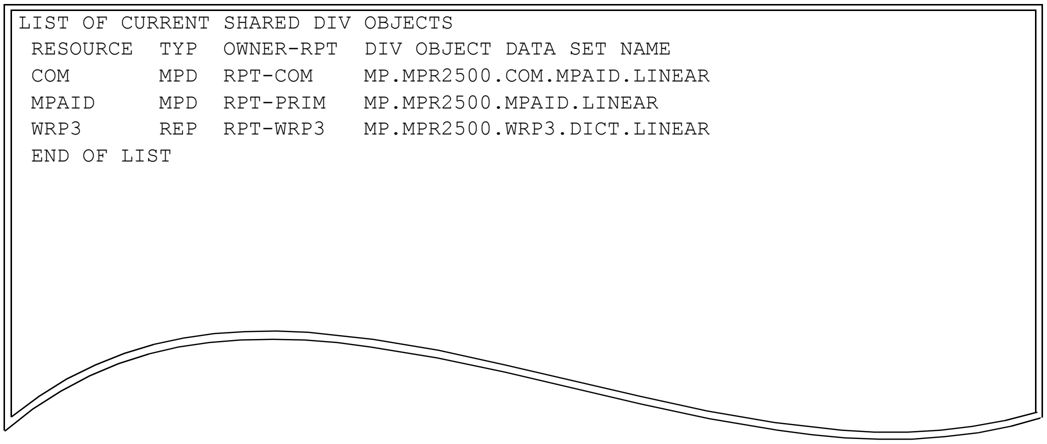


```
SHARED REPOSITORY WRP3 USAGE STATISTICS ON 31 AUG 2000 AT 04.57.12
 TASK          INDEX    SOURCE      DATA   RECOVER       LOG     TOTAL
 RPT-WRP3*
  READS          150        14        20       272       137       593
  WRITES           0         0         0         0         2         2
  TOTAL          150        14        20       272       139       595
 DEVMXC
  READS         7672      4523     82290      1891      1197     97573
  WRITES         618       591      9309      1278      1199     12995
  TOTAL         8290      5114     91599      3169      2396    110568
 SPTFVG
  READS        12568      5527     31930      3597      2392     56014
  WRITES         936       936     10176      2655      2406     17109
  TOTAL        13504      6463     42106      6252      4798     73123
 PROD1500
  READS         8433      3983     21412      2302      1530     37660
  WRITES         543       543      5955      1755      1539     10335
  TOTAL         8976      4526     27367      4057      3069     47995
 OLD-TASKS
  READS        81453     36143    226768     23360     15285    383009
  WRITES        6435      6300     79775     17020     15378    124908
  TOTAL        87888     42443    306543     40380     30663    507917
 ALL-TASKS
  READS       110276     50190    362420     31422     20541    574849
  WRITES        8532      8370    105215     22708     20524    165349
  TOTAL       118808     58560    467635     54130     41065    740198
```

A further form of the QUERY SHARED command provides a list of currently shared DIV resources with the name of their associated VSAM objects (linear datasets). The syntax of the command is:

```
QUERY SHARED OBJECTS;
```

Figure 16 • Example output from the QUERY SHARED OBJECTS command



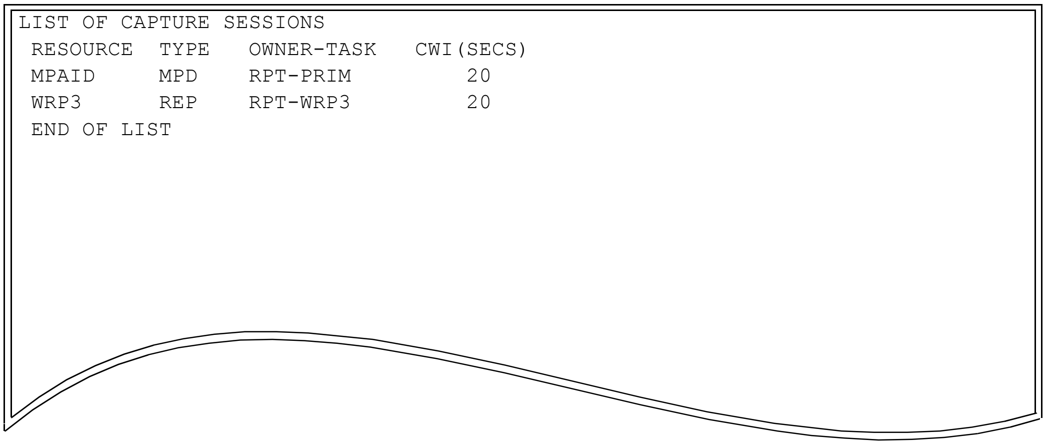
```
LIST OF CURRENT SHARED DIV OBJECTS
 RESOURCE  TYP  OWNER-RPT  DIV OBJECT DATA SET NAME
 COM       MPD  RPT-COM    MP.MPR2500.COM.MPAID.LINEAR
 MPAID     MPD  RPT-PRIM   MP.MPR2500.MPAID.LINEAR
 WRP3      REP  RPT-WRP3   MP.MPR2500.WRP3.DICT.LINEAR
 END OF LIST
```

The QUERY CAPTURE Command

The QUERY CAPTURE command lists all active CAPTURE sessions together with details of the owning RPT and the current capture interval. The syntax of the command is:

```
QUERY CAPTURE;
```

Figure 17 • Example output from the QUERY CAPTURE command



```
LIST OF CAPTURE SESSIONS
 RESOURCE  TYPE   OWNER-TASK   CWI(SECS)
 MPAID     MPD    RPT-PRIM        20
 WRP3      REP    RPT-WRP3        20
 END OF LIST
```

An extended form of the QUERY CAPTURE command provides additional information and statistics for each CAPTURE session. For each CAPTURE session the following output is produced:

Session start	Date and time of CAPTURE session start.
Detection interval	Interval (in seconds) between capture attempts.
Capture attempts count	Total capture attempts this session.
Capture actual count	Actual captures this session.
Capture hit rate	Percentage of capture attempts resulting in update write back.
Capture updates	Total number of updates written back this session.
Average updates/capture	Average number of updates written back per capture.

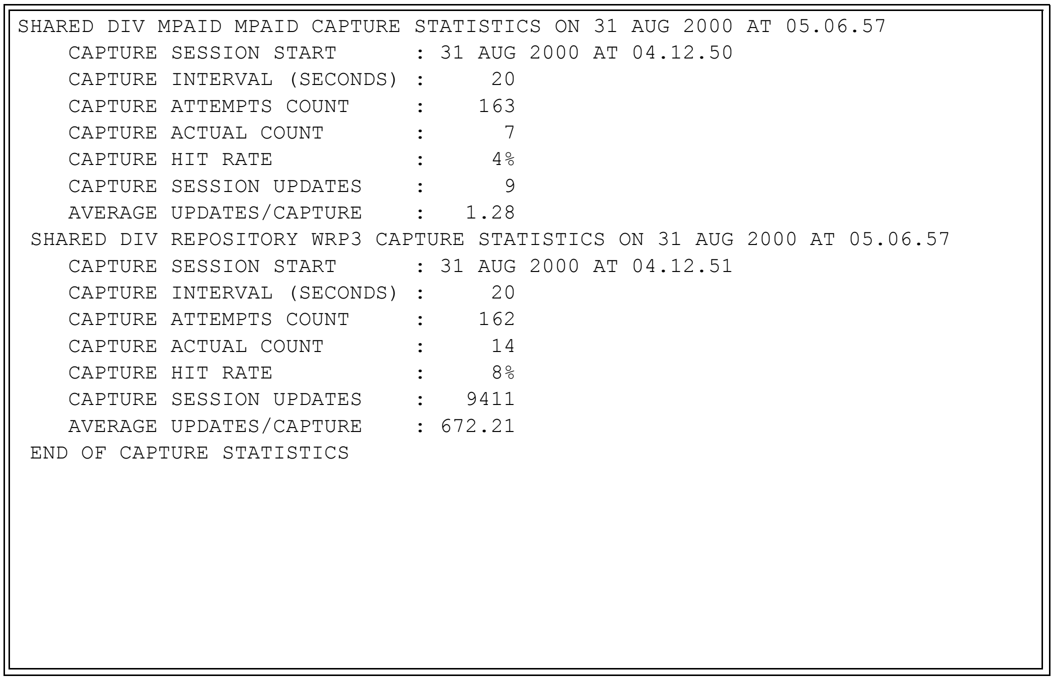
A high average updates/capture count can indicate a need to reduce the update detection interval.

A low capture hit rate can indicate a need to increase the update detection interval.

The syntax of the extended command is:

```
QUERY CAPTURE ALL;
```

Figure 18 • Example output from the QUERY CAPTURE ALL command



```
SHARED DIV MPAID MPAID CAPTURE STATISTICS ON 31 AUG 2000 AT 05.06.57
    CAPTURE SESSION START      : 31 AUG 2000 AT 04.12.50
    CAPTURE INTERVAL (SECONDS) :      20
    CAPTURE ATTEMPTS COUNT     :     163
    CAPTURE ACTUAL COUNT       :       7
    CAPTURE HIT RATE           :      4%
    CAPTURE SESSION UPDATES    :       9
    AVERAGE UPDATES/CAPTURE    :    1.28
 SHARED DIV REPOSITORY WRP3 CAPTURE STATISTICS ON 31 AUG 2000 AT 05.06.57
    CAPTURE SESSION START      : 31 AUG 2000 AT 04.12.51
    CAPTURE INTERVAL (SECONDS) :      20
    CAPTURE ATTEMPTS COUNT     :     162
    CAPTURE ACTUAL COUNT       :      14
    CAPTURE HIT RATE           :      8%
    CAPTURE SESSION UPDATES    :    9411
    AVERAGE UPDATES/CAPTURE    : 672.21
 END OF CAPTURE STATISTICS
```

Optimizing Performance

MPSF can be configured to optimize response times and maximize the use of available resources. Facilities are provided that allow you to:

- Specify the maximum number of conversations that can be executing at any one time (see "Limiting Concurrent Conversations" on page 64).
- Specify a timeout value for online clients, so that MPSF resources are not allocated unnecessarily to inactive conversations for an extended period of time (see "Setting a Timeout Limit" on page 64).
- Control client virtual storage usage, preventing excessive use of system resources (see "Controlling Client Virtual Storage Usage" on page 66).
- Make the server non-swappable, avoiding swap-out and swap-in overheads (see "Making the Server Non-swappable" on page 67).

Limiting Concurrent Conversations

When you create your subtasking environment (see "Creating the Subtasks" on page 36), you define sufficient subtasks for the maximum number of users requesting concurrent access to Manager Products resources. The number of subtasks is fixed before MPSF is started and cannot be dynamically increased. You can use the QUERY SERVER command to determine the current conversation limit.

However, there may be times when you want to limit the number of concurrent conversations that can be allocated in order to reduce server load. This can improve response time for currently allocated conversations, particularly when the repository is being updated extensively.

To set a limit on the number of concurrent conversations, issue this command to the QPT:

```
SUBTASK QSTART nn 'SERVER MAX-CONV xx';
```

To remove a previously set limit and reset the number of concurrent conversations to that defined at the session start, issue this command:

```
SUBTASK QSTART nn 'SERVER MAX-CONV OFF';
```

where:

nn is the number or name of the QPT subtask.

xx must be an integer between 1 and the number of subtasks defined minus 1 that represents the maximum number of concurrent conversations you wish to allocate.

For example, this command sets the concurrent conversation limit to 50: `SUBTASK QSTART 1 'SERVER MAX-CONV 50';`

Setting a Timeout Limit

A Client Processing Task (CPT) is created each time a conversation is successfully allocated. The CPT executes in a subtask allocated from the pool of available subtasks. The CPT processes each client request, returns any generated output, and waits for further requests from the client. If the client side of the conversation fails to respond or is terminated for any reason (that does not cause normal or abnormal termination of the entire conversation), then one or more CPTs can be left waiting for client input for an indefinite period of time using resources that might otherwise be available for new conversations.

MPSF terminates any waiting CPT and its APPC conversation or TCP/IP connection when its timeout value is exceeded. The client is notified that a timeout has occurred whenever possible. The timeout facility allows you to set a value globally for all clients or locally for a particular user.

Changing the Global Value

Determine your current global timeout value with the QUERY SERVER command (see "The QUERY SERVER Command" on page 51). The default global value is 30 minutes.

To change or disable the global timeout value, issue this command to the QPT:

```
SUBTASK QSTART nn 'SERVER CLIENT-TIME-OUT xx';
```

where:

nn is the number or name of the QPT subtask.

xx is one of these values:

— An integer from 1 to 1440 (24 hours) that represents the timeout value in minutes. If a number greater than 1440 is specified, 1440 is assumed.

— `OFF` to specify that the timeout facility is disabled at the global level.

`CLIENT-TIME-OUT` can be replaced with the initials `CTO.`

This command can be issued at any time during MPSF execution. Message 14544I is sent to the MPSF log if the command is successful. If the command is not successful, message 14545E is sent.

For example, this command sets the global timeout value to 20 minutes: `SUBTASK QSTART 1 'SERVER CTO 20';`

Changing a Local Value

You can change a local value for a particular Manager Products user. A local value takes precedence over any global value.

The command to change or disable a local value is usually added to the user logon profile and executed as part of a Manager Products logon procedure. This is the command:

```
SET CLIENT-TIME-OUT  xx;
```

where *xx* is one of these values:

— An integer from 1 to 1440 (24 hours) that represents the timeout value in minutes. If a number greater than 1440 is specified, 1440 is assumed.

— `ON` to remove any local value and re-enable any global value.

— `OFF` to specify that the timeout facility is disabled for this user even if a global value is set.

Any user can determine their own timeout value by issuing this command:

```
QUERY CLIENT-TIME-OUT;
```

where `CLIENT-TIME-OUT` can be replaced with the initials `CTO`.

For example, this command sets a local timeout value of one hour: `SET CLIENT-TIME-OUT 60;`

Controlling Client Virtual Storage Usage

In MPSF environments all clients execute as OS/390 subtasks in a single OS/390 address space and all client requests for virtual storage are satisfied from the private space available to the address space. Uncontrolled availability and usage of virtual storage can have a serious impact on MPSF performance and in extreme circumstances could lead to a complete failure if insufficient storage is available to the MPSF management tasks when requested.

MPSF provides a means of restricting the total amount of virtual storage that can be requested by a client. A limit can be set for individual clients or groups of clients, depending on their use of Manager Products functionality.

The Systems Administrator issues a command to set a limit which is normally added to the Logon Profile of the user.

To set a limit enter:

```
SET  VIRTUAL-STORAGE-MAXIMUM  nnnnnn ;
```

where *nnnnnn* is the total amount of virtual storage, in K, that is allocated to the user.

You can specify a maximum size of 999999 or a minimum size of 1000. If the current virtual storage used exceeds the specified limit, the current value is used. ASG recommends a starting value of 5000, which can be adjusted if necessary.

Users can query their limit by entering:

```
QUERY VIRTUAL-STORAGE-MAXIMUM ;
```

VSM is an acceptable short form of VIRTUAL-STORAGE-MAXIMUM.

You can also obtain a snapshot of current virtual storage usage for all MPSF tasks by entering:

```
QUERY SERVER STORAGE ;
```

Refer to "Monitoring Virtual Storage Usage and Availability" on page 56 for an explanation of the output from this command.

Making the Server Non-swappable

You can enhance MPSF performance and response times by designating the address space in which MPSF executes as non-swappable. When non-swappable, the overheads associated with swap-out and subsequent swap-in of the address space are avoided.

Note:

Before enabling this MPSF feature, you should inform your Systems Programming Group of your requirement to run MPSF in non-swappable mode.

To make the MPSF address space non-swappable, use the NON-SWAPPABLE keyword of the SERVER START command. Refer to "Activating MPSF" on page 44 for a description of the SERVER START command.

Suspending and Resuming the Server

There may be times you wish to suspend execution of the QPT without completely terminating MPSF. For example, if a high number of active conversations has increased response time, you may want to prevent further conversations from starting until the server load is reduced.

Send this command to the QPT to suspend MPSF:

```
SUBTASK QSTART nn 'SERVER SUSPEND';
```

where *nn* is the number or name of the QPT subtask.

When MPSF has been suspended successfully, message DM14501I is sent to the MPSF log and no further conversations are allocated. Existing conversations continue to execute normally.

Note:

If you start a conversation in order to suspend MPSF, make sure your conversation does not terminate until you have resumed MPSF. If your conversation terminates, you will not be able to start a new conversation so that you can resume MPSF.

Send this SERVER RESUME command when you wish to resume execution of the QPT:

```
SUBTASK QSTART nn 'SERVER RESUME';
```

where *nn* is the number or name of the QPT subtask.

When MPSF has been resumed successfully, message DM14502I is sent to the MPSF log.

Terminating a Conversation

You may need to terminate a conversation for any of several reasons, such as a loop condition causing excessive CPU usage or an incorrectly submitted command or job that must be purged. Terminate a conversation by cancelling the subtask that is executing as the CPT. You can find the subtask number or name by querying the server (see "The QUERY SERVER Command" on page 51).

Issue this command to the QPT:

```
SUBTASK QSTART nn 'SUBTASK CANCEL xx';
```

where:

nn is the number or name of the QPT subtask.

xx is the subtask number or name of the conversation to be terminated.

The client side of the conversation is notified of termination. No dump is produced. When the conversation has been terminated successfully, message DM14517I is sent to the MPSF log.

Terminating the Server

Sometimes it is necessary to terminate the execution of MPSF. For example, you may want to increase the region size where MPSF executes, reorganize a repository, or perform system maintenance external to Manager Products. You can terminate immediately without waiting for current conversations to end, or you can terminate after all current MPSF conversations have ended. When MPSF has terminated successfully, message DM14520I is sent to the MPSF log. The MPSF environment no longer exists, but the MPSF log is still available for inspection. Send the appropriate termination command when logged on as an MPSF client.

You must then terminate your RPT substasks (see "Terminating the RPT Subtasks" on page 70) before issuing the final LOGOFF command to end execution of Manager Products.

Immediate Termination

To terminate MPSF immediately, without waiting for current conversations to end, send this command to the QPT:

```
SUBTASK QSTART nn 'SERVER END IMMEDIATE';
```

where *nn* is the number or name of the QPT subtask.

Where possible, clients are notified of server-side cancellation.

Termination After Current Conversations End (Non-immediate)

To terminate MPSF after all current conversations have ended, send this command to the QPT:

```
SUBTASK QSTART nn 'SERVER END';
```

where *nn* is the number or name of the QPT subtask.

QPT performs these actions when it receives the SERVER END command:

- Sends a shut-down message to all full-screen TSO clients requesting them to end their Manager Products session.
- Enters the shutdown phase of execution, which does not allow new conversations to begin. When this phase has begun, message DM14519I is sent to the MPSF log.

  MPSF waits for all current conversations to end (unless termination is immediate). Message DM14532I is sent to the MPSF log every 60 seconds, showing the number of conversations still active.
- De-registers MPSF to APPC/MVS.
- Terminates the TCP/IP interface, if started.
- Releases all virtual storage acquired for the MPSF environment.
- Terminates the QPT.

A successful non-immediate shutdown request can be followed by a later immediate request, if needed.

Changing the Server Shutdown Message

When a non-immediate server shutdown is requested, MPSF writes a shutdown message to all currently logged on full-screen TSO users requesting them to log off from Manager Products. The default message sent is:

```
MPSF shutdown in progress, please LOGOFF now
```

You can provide your own shutdown message text when starting MPSF or at any time during MPSF execution.

Use the SHUTDOWN-MESSAGE clause of the SERVER START command to provide a shutdown message when starting MPSF. Refer to "Activating MPSF" on page 44 for a complete definition of the SERVER START command.

The SERVER SHUTDOWN-MESSAGE command is sent to the MPSF QPT to change the current shutdown message at any time during MPSF execution.

Send this command to the QPT to specify a new shutdown message:

```
    SUBTASK QSTART nn 'SERVER SHUTDOWN-MESSAGE "shutdown-message" ' ;
```

where:

nn is the number or name of the QPT subtask. The maximum length of a shutdown message is 70 characters.

shutdown-message is the message you want to display.

Execute a QUERY SERVER command to determine the current shutdown message.

Terminating the RPT Subtasks

Termination of your RPT subtasks is normally performed within your MPSF startup batch job after MPSF has been terminated using the SERVER END command issued from an MPSF client. You need to issue a CAPTURE STOP command to the RPT, if a CAPTURE session is currently active, followed by a LOGOFF command to end the RPT subtask. You can also print the content of the RPT buffer by using a SUBTASK VIEW command.

You must terminate your RPT subtasks in the following order, repositories first, followed by your primary MPAID and ending with your secondary MPAIDs. If the order is incorrect, termination will fail if you attempt to shutdown an RPT where the owned DIV resource is still in use by another RPT.

The following commands should be issued to terminate an RPT subtask:

```
SUBTASK START nn 'CAPTURE STOP';
SUBTASK WAIT nn;
SUBTASK VIEW nn;
SUBTASK START nn 'LOGOFF';
SUBTASK WAIT;
```

where *nn* is the number or name of the RPT subtask.

You should omit the first two commands when no CAPTURE session is active.

If you want to shutdown an RPT subtask during MPSF execution then you must replace the START/VIEW keywords with RSTART/RVIEW. You can only terminate the RPT when no other users are accessing (i.e., have open) the DIV resource owned by the RPT. If there are still users accessing the resource, any LOGOFF command issued to the RPT is refused and the DIV resource remains available to the current users. However, new requests to use the resource are refused as the resource is considered to be in shutdown mode.

See Appendix A,"Example Batch Start" on page 93 for examples of the Manager Products commands used when terminating RPT subtasks.

Access and Sharing of DIV Resources

Accessing DIV Resources

The Resource Processing Tasks (RPTs) of MPSF make available your DIV resources (repositories and MPAIDs) by opening them in shared mode at server initialization time.

A DIV resource allocated and available under a specific MPSF execution cannot be allocated under *any* other MPSF execution, except for MPAIDs, used as read-only concatenations, which can be shared across multiple MPSF instances. See "Sharing Read-only Concatenated MPAIDs" on page 75 for details.

DIV resources are only available to Manager Products users executing as MPSF clients.

OS/390 clients (e.g., batch and TSO) must execute under the same OS/390 image as the MPSF server job or started task unless cross-system support in enabled. See "Configuring APPC and VTAM" on page 33 for more details.

Shared Usage of DIV Resources

DIV resources can be shared between Manager Products users in the same way as BDAM or VSAM resources can be shared. Both Standard and Alternative Usage for repositories are supported. Refer to *ASG-Manager Products Installation in OS Environments* for a detailed discussion of Standard and Alternative Usage.

When using BDAM or VSAM resources protection against concurrent updating is provided by use of ENQ/DEQ SVCs. Protection for DIV resources is provided by utilizing MPSF virtual storage locks. Processor instructions are used to obtain shared or exclusive control of the appropriate lock for the requested DIV resource.

Up to four locks per resource are employed:

The SERIALIZATION lock. Exclusive control of this lock is first obtained by all users requiring read or write access to a repository resource and is used to ensure that multiple users requiring read-only access to the resource are not permitted to monopolize the resource for long periods of time.

The EXCLUSIVE lock. Exclusive control of this lock is obtained by all users requiring write access to the resource. Once obtained by a user, no other user, whether requiring read or write access, can use the resource. Control of this lock can only be obtained when the lock is available *and* the SHARED lock usage count is zero.

The SHARED lock. Shared control of this lock is obtained by all users requiring read access to the resource. Once obtained by a user, only other users requiring read access, can use the resource. A user count is incremented when a user obtains shared control and is decremented when control is relinquished. Shared control of this lock can only be obtained when the EXCLUSIVE lock is available.

The LOG lock. Exclusive control of this lock is obtained by all users requiring write access to the LOG component of a repository resource.

When accessing a DIV MPAID, the SERIALIZATION and LOG locks are not used.

For all of the above lock types, when control of the required lock cannot be obtained, the user task is placed in a wait state until a timer interval expires, at which point a further attempt is made to obtain control of the lock. This process is repeated until control is obtained. You can control the timer interval value for each user using the SET BUSY-WAIT-INTERVAL command. The default value used is 20 milliseconds (1/50 of a second).

In situations where many MPSF users compete for access to the same resource, you can use this command to alter the priority at which they execute when obtaining locks. For example, you can give your TSO users higher priority than batch users by setting a higher timer interval value for those batch users than the TSO users. Batch users will wait for a longer period than TSO users when unable to obtain a lock and thus enable greater TSO throughput.

You may also need to adjust the timer interval value according to the speed of your processor and Manager Products workload. For example, if users are accessing the repository with Alternative Usage, recommended for large repositories accessed by many users, then for maximum performance you should set the timer interval value to half the average elapsed time of your repository updating commands. You can obtain accurate elapsed time information from the repository log using the AUDIT command.

To set a new timer interval value, enter this command:

```
SET BUSY-WAIT-INTERVAL nn ;
```

where *nn* is the new value in milliseconds and must be in the range 1 to 5000.

The command is restricted to execution by the Systems Administrator.

A valid short form of the BUSY-WAIT-INTERVAL keyword is BWI.

All users can query their current setting using:

```
QUERY BUSY-WAIT-INTERVAL;
```

Or

```
QUERY BWI;
```

Monitoring Locks and Lock Waits

MPSF provides commands that allow you to monitor lock and lock wait activity for your DIV resources. Information is provided for current and completed MPSF tasks to help you determine an optimum BUSY-WAIT-INTERVAL for MPSF tasks accessing DIV resources and identify throughput bottlenecks. You should investigate resources with high Wait Averages. The resource may be heavily utilized or there may be some underlying problem that can be resolved (such as performing repository or MPAID reorganization).

To obtain information for all currently shared DIV resources enter:

```
    QUERY SHARED LOCKS ;
```

To obtain information for a specific DIV resource enter:

```
    QUERY SHARED LOCKS resource-name ;
```

where *resource-name* is the name of a currently shared DIV repository or MPAID.

This information is output for each DIV resource:

Task Identity Name of task using resource. For a QPT or RPT this is a subtask name and for a CPT it is the client identity passed at client connection time. OLD-TASKS and ALL-TASKS identify task summary outputs. An asterisk (*) to the right of the task name indicates the owning RPT for the resource.

Task Mode Task execution mode. For MPSF clients the values shown are the same as that of the QUERY SERVER output. QPT and RPT are output for MPSF management tasks.

Task BWI Task current Busy-Wait-Interval, in milliseconds.

Lock Total Number of read, write, and total locks requested by task.

Wait Total Number of read, write, and total lock waits needed to service lock requests.

Wait Average Average number of waits needed to service lock requests.

Figure 19 • Output from the QUERY SHARED LOCKS command

```
SHARED REPOSITORY SAMP LOCK STATISTICS ON 21 SEP 2001 AT 03.42.52
 TASK          TASK    TASK     LOCK      WAIT     WAIT
               MODE    BWI      TOTAL     TOTAL    AVERAGE
 RPT-SAMP*     RPT       20
  READ                               1176      2004       1.70
  WRITE                                84         0       0.00
  TOTAL                              1260      2004       1.59
 DEVMXC        TSO       20
  READ                                 11         0       0.00
  WRITE                               593       154       0.25
  TOTAL                               604       154       0.25
 DEVMXC45      BAT       30
  READ                                  3         0       0.00
  WRITE                              1001       571       0.57
  TOTAL                              1004       571       0.56
 DEVMXC75      BAT       30
  READ                                  3         3       1.00
  WRITE                               734       602       0.82
  TOTAL                               737       605       0.82
 OLD-TASKS
  READ                                 30       100       3.33
  WRITE                             10023      4724       0.47
  TOTAL                             10053      4824       0.47
 ALL-TASKS
  READ                               1223      2107       1.72
  WRITE                             12435      6051       0.48
  TOTAL                             13658      8158       0.59
```

Displaying and Resetting DIV Locks

Locks obtained for a DIV resource are automatically released when the command accessing the resource terminates. This is also true for most abnormal terminations.

However, there may be isolated situations where one or more locks are not released, for example, after a system abend not trapped by Manager Products. In these situations, further usage of the DIV resource is not possible and queuing MPSF clients will continue to wait.

You can determine if such a situation has occurred by using the Systems Administrator restricted QUERY LOCKS command. Output from this command can be used to determine if locks are permanently held by a Manager Products task that has previously terminated abnormally. Normally lock information is constantly changing as locks are obtained and released by MPSF clients. A lock held for an extended period by the same client indicates a potential problem that may need investigation.

The following is an example of the output from a QUERY LOCKS command:

```
LIST OF CURRENT LOCKS
 RESOURCE  TYP  LOCK-TYPE  LOCK-OWNER-INFORMATION
 PROD      REP  EXCLUSIVE  TSKNO=0012 JOBNO=0047 JOBID=DEVMXC20
                SERIAL     TSKNO=0008 JOBNO=0043 JOBID=DEVMXC10
 END OF LIST
```

The example shows the resource name and type, locks held together with information about the MPSF client owning the lock.

You can explicitly release the locks held for a specific DIV resource by issuing the Systems Administrator restricted command:

```
RESET RESOURCE resource-name;
```

where *resource-name* is the name of a DIV resource and must be from 1 to 8 characters in length.

Sharing Read-only Concatenated MPAIDs

When running more than one instance of MPSF you can share one or more read-only, concatenated (secondary) MPAIDs between your servers. Disk space usage and maintenance efforts are kept to a minimum by keeping a single copy of your read-only MPAID members.

To share secondary MPAIDs successfully

1 Create the MPAID VSAM cluster with SHAREOPTIONS (2,3).

This permits the cluster to be as shared read-only by any number of OS/390 address spaces and updated by just one.

2 Allocate the JCL DD statement in the MPSF startup jobs using DISP=SHR.

If you need to update a secondary MPAID, it is possible to open it as a primary MPAID in update mode from one MPSF instance only. You can then perform your updates, but the changes made will *not* necessarily be seen by other MPSF clients executing in other MPSF instances and having the updated MPAID accessed as a secondary MPAID.

To make your changes visible to other MPSF instances, issue this command to all RPTs owning the updated MPAID as a secondary MPAID:

```
SUBTASK RSTART nn 'DIV REFRESH' ;
```

where `nn` is the number or name of the RPT subtask.

Sharing violations, that is, attempting to open an MPAID in update mode from more than a single MPSF instance, are handled by OS/390 and if detected result in the output of this message: `'IEC161I 052(013) - 84, user variable information'`.

5 The Server Log

This chapter discusses the server log and describes the general messages sent to the log by MPSF, and contains these sections:

Overview

The QPT creates a virtual storage log during MPSF execution and contains messages for each allocated or rejected conversation. There are some other events that result in informational messages to the log:

- Starting MPSF
- Altering MPSF Concurrent Conversation Limit
- Altering MPSF Global Timeout Value
- Terminating MPSF
- Suspending MPSF
- Resuming MPSF
- Changing the Date

Output from all commands sent to the QPT is also written to the log.

Viewing the Log

Use this SUBTASK QVIEW command to view the log at any time during MPSF execution:

```
SUBTASK QVIEW nn;
```

where *nn* is the number or name of the QPT subtask.

Once MPSF is terminated, you can view the log until the subtask used for the QPT is terminated or used for another purpose.

Conversation Start Messages

Each time QPT accepts a conversation start, it issues message DM14511I to the server log. This is the information in that message:

Field	Description
Client Identity	Identity of the client holding a conversation with Manager Products. Can be the batch job name, TSO logon ID, etc.
Conversation Number	The four-digit conversation number allocated by MPSF. This number uniquely identifies the conversation if the client's identity is duplicated.
Connection Time	The time when the conversation began.
Client Count	A concurrent conversation count.
Client Mode	The execution mode of the client. These are the values: A Access Call B Batch F Full-Screen TSO I TSO/ISPF (SU FE70) M ManagerView P PWS based client R ROSCOE W Web Enabler X REXX Interface

Conversation Termination Messages

The QPT sends a message whenever a conversation is terminated. These are the five possible termination messages sent to the server log:

Message	Description
DM14512W	Normal termination has occurred.
DM14513W	Client Processing Task (CPT) abended.
DM14514W	Client timeout occurred.
DM14515W	Client side has terminated.
DM14516W	Client Processing Task (CPT) cancelled.

This is the information in each message:

Field	Description
Client Identity	Identity of the client holding a conversation with Manager Products. Can be the batch job name, TSO logon ID, etc.
Conversation Number	The four-digit conversation number allocated by MPSF.
End Time	The time when the conversation terminated.
Client Count	A concurrent conversation count.
Return Code	The Manager Products return code.
CPU Time	CPU time used (in seconds to two decimal places).

Abnormal termination messages include information identifying the reason for the abend.

Rejected Conversations

When MPSF rejects a conversation, message DM14515W is sent to the server log. That message includes this information:

Field	Description
Client Identity	Identity of the client holding a conversation with Manager Products. Can be the batch job name, TSO logon ID, etc.
Conversation Number	The conversation number set to zero (0000) by MPSF.
Rejection Time	The time when the conversation was rejected.
Client Count	A concurrent conversation count.
Reason Code	One of these reason for rejection codes: 4 Server has no free subtasks. 8 Server is currently suspended. 14 Client/server software is incompatible. 28 Server has reached its maximum conversation limit.

6 Miscellaneous Server Facilities

You can utilize a number of optional facilities to enhance the usability of MPSF. Optional facilities can include:

Writing System Management Facilities (SMF) Records

An optional facility of MPSF supports the writing of SMF records during execution of the server. When enabled, a single record is written for each MPSF client at termination of the client conversation. You can use these generated records to analyze usage of MPSF and also as a basis for user charging.

The size of each record is 132 bytes and consists of a standard SMF record header followed by a number of MPSF specific fields, the contents of which are shown in the table on page 82.

Use the SMF-RECORD-TYPE keyword of the SERVER START command to enable generation of SMF records. You must supply an SMF record type to be written in the range of 120 to 255. No default record type is provided. Refer to "Activating MPSF" on page 44 for further details of the SMF-RECORD-TYPE clause. If an invalid record type is specified, message 14560E is output and MPSF execution terminates.

Consult your Systems Programming Group to determine a suitable record type and if necessary update the SMF execution parameters to ensure that the chosen record type is selected for writing to the SMF dataset. You can achieve this by using the TYPE option of the SYS or SUBSYS parameter of the appropriate SMFPRM*xx* member of SYS1.PARMLIB or by updating the appropriate installation-written exit routines IEFU*xx*. For further information consult your IBM SMF documentation.

If, during the course of MPSF execution, a failure occurs when writing SMF records, message 14561W is output giving the reason for the failure, followed by message 14562. Writing of SMF records is then terminated but MPSF execution continues as normal.

Table • Description of MPSF SMF Record

ASG Field Name	Dec Offset	Hex Offset	Format	Length	Description
SMFLL	00	00	Binary	2	Record length. Set to X'0084' (decimal 132)
SMFBB	02	02	Binary	2	Unused, set to X'0000'
SMFIND	04	04	Binary	1	System indicator. Set to X'1E'
SMFRT	05	05	Binary	1	SMF record type. As specified by user in range X'80' to X'FF' (decimal 128 to 255)
SMFTIME	06	06	Binary	4	Time since midnight, in hundredths of a second, when record was written to the SMF buffer
SMFDATE	10	0A	Packed	4	Date when record was written to the SMF buffer, in the form 01*yydddF* (where *yy* is the current year (00-99), *ddd* is the current day (001-366) and *F* is the sign
SMFID	14	0E	EBCDIC	4	System identification (as specified by SID parameter of SMFPRMxx member)
SMFSVRN	18	12	EBCDIC	8	MPSF server name (blank padded)
SMFCID	26	1A	EBCDIC	8	Client identity (blank padded)

ASG Field Name	Dec Offset	Hex Offset	Format	Length	Description
SMFUID	34	22	EBCDIC	8	Client RACF/ACF2 user ID (blank padded)
SMFMODE	42	2A	EBCDIC	1	Client execution mode. Set to: A for Access Call B for Batch F for Full-screen TSO I for TSO/ISPF M for ManagerView P for PWS based client R for Roscoe W for Web Enabler X for REXX interface
SMFRETC	43	2B	Binary	1	Manager Products return code
SMFJNO	44	2C	Binary	4	Client job number (1-9999)
SMFCPU	48	30	Binary	4	CPU time used by client, in hundredths of a second
SMFCSD	52	34	Packed	4	Date of client conversation start, in the form: 01*yydddF* (where *yy* is the current year (00-99), *ddd* is the current day (001-366) and *F* is the sign
SMFCST	56	38	Binary	4	Time since midnight, in hundredths of a second, of client conversation start
SMFCED	60	3C	Packed	4	Date of client conversation end, in the form: 01*yydddF* where *yy* is the current year (00-99), *ddd* is the current day (001-366) and *F* is the sign
SMFCET	64	40	Binary	4	Time since midnight, in hundredths of a second, of client conversation end

ASG Field Name	Dec Offset	Hex Offset	Format	Length	Description
SMFMSGIN	68	44	Binary	4	Client input message count
SMFCTR	72	48	Binary	4	Client total response time, in hundredths of a second. This value, divided by the input message count gives the average MPSF response time
SMFDIN	76	4C	Binary	4	Client data received (in K)
SMFDOUT	80	50	Binary	4	Client data sent (in K)
SMFABD	84	54	EBCDIC	4	Client abnormal termination code (normally set to binary zeroes)
SMFRSVD	88	58	Binary	44	Reserved by MPSF. Set to binary zeroes

Manager Diagnostic Facility

The Manager Diagnostic Facility (MDF) supports the storage and subsequent retrieval of diagnostic dumps that can be produced during the execution of Manager Products by MPSF clients. When enabled, dumps produced from any Manager Products internal errors or program checks are stored in virtual storage where they can be viewed and manipulated in various ways by the Systems Administrator. Other system abends are not intercepted by Manager Products and are written directly to SYSUDUMP. The facility provides an easy method of tracking problems that may be encountered by users. Any duplicate dumps can be discarded thus reducing to a minimum the amount of diagnostic data that may need to be sent to ASG. If MDF is not enabled, all diagnostic dumps are written to the Manager Products MPRDIAG dataset.

Enable MDF by specifying the MDF keyword when starting MPSF using the SERVER command. A default 31-bit buffer of 20MB is allocated. You can change the buffer size at any time during MPSF execution.

Listing MDF Diagnostic Dumps

Use the MDF LIST command to obtain a list of any dumps written to the MDF buffer. The keyword LIST is defaulted and can be omitted if required.

Each item in the list consists of an identifier, brief problem description, line count, and creation date and time. The buffer size in MB and percentage utilization is also provided.

The dump identifier is a concatenation of the client identity and 4 digit conversation number.

A line count shown as `******` indicates that no space was available to begin writing the dump.

A line count suffixed with an asterisk (`*`) indicates that the dump was not completed due to insufficient space.

In both cases you must delete some unwanted entries or write out some entries to an external dataset prior to deleting them. You should also consider allocating a larger buffer.

Figure 1 • Example of output from the MDF LIST command

```
LIST OF CURRENT MPR DIAGNOSTIC OUTPUT
 IDENTIFIER   DESCRIPTION                       LINES  CREATION DATE/TIME

 DEVMXC0002   921 ABEND/MOD=DMI01/CMD=LIST ONLY  7383  27 SEP 2000/05.59.50

 MDF BUFFER SIZE   :  20MB
 MDF BUFFER USAGE :    4%
```

Viewing MDF Diagnostic Dumps

Print or view on the screen the content of any diagnostic dump using the command:

```
MDF PRINT identifier;
```

where *`identifier`* is a dump identifier as shown in the output of an MDF LIST command.

Deleting MDF Diagnostic Dumps

Delete any diagnostic dump by using the command:

```
MDF DELETE identifier;
```

where *`identifier`* is a dump identifier as shown in the output of an MDF LIST command.

Delete all the diagnostic dumps by using the command:

```
MDF DELETE ALL;
```

Writing MDF Diagnostic Dumps to an External Dataset

Write any diagnostic dump to an external dataset by using the command:

```
MDF WRITE identifier <ONTO ddname> ;
```

where:

> *identifier* is a dump identifier as shown in the output of an MDF LIST command.
>
> *ddname* specifies the DDNAME of the output dataset and is a maximum of 8 characters in length.

If the ONTO clause is omitted then a DDNAME of MPRDIAG is used.

Attributes of the output dataset are:

```
RECFM=FBA
LRECL=133
BLKSIZE=13300
```

A diagnostic dump written to an external dataset is retained in the MDF buffer until deleted by a subsequent MDF DELETE command.

Changing the Size of the MDF Buffer

Change the size of the MDF buffer at any time during MPSF execution by using the command:

```
MDF BUFFER-SIZE nn;
```

where *nn* is the new size (in MB) of the buffer and must be in the range 1 to 200.

If a new MDF buffer is successfully allocated then the old buffer is deleted along with any previous content.

The MDF buffer is owned by the MPSF Queue Processing Task (QPT) and can only be reallocated by that task. You must therefore send the MDF BUFFER-SIZE command to the QPT for execution using the SUBTASK QSTART command.

Server Message Facility

The Server Message Facility (SMF) allows the Systems Administrator to specify and store a number of message strings for subsequent retrieval by any MPSF client. MPSF clients retrieve the stored strings using the Procedures Language GETSVRM function.

The facility provides a simple and convenient way for you to distribute common information to Manager Products users. Up to 50 message strings can be defined.

Setting up the Messages

Message strings are defined using the command:

```
SET SERVER-MESSAGE nn 'message-string' ;
```

where:

nn is an integer between 1 and 50 specifying the number of the message string to be defined.

message-string is any text up to 255 characters in length.

Remove a previously defined message string using the command:

```
SET SERVER-MESSAGE nn OFF;
```

Retrieving the Messages

Message strings are retrieved using the Procedures Language GETSVRM function.

This example is a simple executive which displays any defined messages:

```
mpxx
do for 50
 if getsvrm(fdo(dfor)) eq '' then iterate
 say getsvrm(fdo(dfor))
 end
```

Dataset Services Facility

The Dataset Services Facility (DSS) provides support for the dynamic allocation and de-allocation of existing sequential and partitioned datasets in the server address space while executing Manager Products as an MPSF client.

You can also allocate the datasets of a BDAM or VSAM repository and MPAID. You cannot currently allocate a dataset for a DIV repository or MPAID.

If you use the Manager Products ISPF/ISPEXEC command to allocate a dataset, then that allocation is made in the client address space and *not* in the server address space. Usage of that dataset by Manager Products functionality executing in the server address space imposes an overhead caused by the need to transmit the data from/to the client address space using APPC communications. Systems Administrator tasks, such as repository or MPAID backups, are performed much faster and efficiently by having the required external datasets allocated in the server address space.

Further, allocation in the client address space restricts usage of that dataset to the client performing the allocation and prevents sharing among all MPSF clients. For example, you may want to allocate a new partitioned dataset to be used by all clients when creating output from the Source Language Generation facility.

The DSS facility is invoked using the DSS command which is restricted to use by the Systems Administrator.

Allocating a Dataset

Allocate an existing dataset using the command:

```
DSS ALLOC

    DD    ddname

    DSN   data-set-name

    DISP  disp

    DSORG dsorg

    FREE  free-opt

    MSG   msg-opt

    ;
```

where:

ddname is a 1 to 8 character DDNAME.

data-set-name is a 1 to 44 character physical dataset name.

disp is the required disposition of the dataset, either SHR or OLD.

dsorg is the dataset access method, either PS for a sequential dataset or PO for a partitioned dataset.

free-opt specifies whether the dataset is to be freed (de-allocated) at close time (YES) or is to remain allocated (NO).

msg-opt specifies how allocation messages are to be processed:

NONE	suppress all messages
ERROR	only if an error occurs
ALL	output all messages

If the DISP clause is omitted SHR is assumed.

If the DSORG clause is omitted PS assumed.

If the FREE clause is omitted NO is assumed.

If the MSG clause is omitted ALL is assumed.

Manager Products BDAM or VSAM repositories and MPAIDs must be allocated as sequential datasets (defaulted or specified as DSORG PS).

Examples:

```
DSS ALLOC DD SYS1 DSN SYS1.MACLIB DSORG PO;
```

```
DSS ALLOC DD PRODV DSN MPR.REL2500.PROD.DICT.SAVE;
```

De-allocating a Dataset

De-allocate (free) a currently allocated dataset using the command:

```
DSS   FREE
      DD   ddname
      MSG  msg-opt
      ;
```

where:

`ddname` is the 1 to 8 character DDNAME of the dataset to be de-allocated.

`msg-opt` specifies how allocation messages are to be processed:

NONE	suppress all messages
ERROR	only if an error occurs
ALL	output all messages

Example:

```
DSS FREE DD SYS1;
```

Additional Dataset Services

There are a number of additional services that can be performed using the DSS command. These include:

- Allocation of new datasets
- Extraction of dataset attributes
- Display of partitioned dataset directory
- Display of partitioned dataset members
- Dataset copy
- Dataset compare
- Dataset delete
- Find dataset or partitioned member
- Display log of session allocations/deallocations

These features are not currently documented in this publication. Use the DSS ? command to obtain details of the command syntax.

7 Server Compatibility

This chapter describes the compatibility issues between Manager Products when implemented to execute under MPSF and when implemented for non-MSPF execution.

Basically the functionality Manager Products provides is equally available whether you are executing under MPSF or not. However, some incompatibilities and limitations when executing under MPSF do exist and are described below. As usage of MPSF continues by ASG developers and the customer base further incompatibilities and restrictions may be encountered.

These sections are included:

Modifying the Manager Products Program Library

For execution under MPSF, Manager Products must run from an APF-authorized library. You must therefore ensure that any changes you make to the user customizable ASG-supplied load modules, such as the Logon exit, MPLX1 and User Defined functions, MPLUF, are link-edited correctly using module attributes that accurately reflect the changes you make. For example, you cannot link-edit a module as RENT if the changes you make contain non-reentrant code. Failure to link-edit modules correctly will result in abnormal terminations (normally an 0C4 abend) when running Manager Products. For non-MPSF execution from a non-APF-authorized library such abnormal terminations might not occur.

You will need to restart MPSF after making any changes to the Manager Products Program Library. This applies also when applying ASG-supplied fixes to the library.

User Subtask Facility

You cannot execute the ISPF/ISPEXEC commands when sent to a background subtask, (i.e., using the SUBTASK START or SUBTASK SEND commands). They are accepted when issued from a foreground subtask, (i.e., a subtask executing in full-screen mode entered using a SUBTASK SWAP or SUBTASK ROTATE command).

In addition, you cannot read from or write to datasets allocated on the client side in the same circumstances as described above for the ISPF/ISPEXEC commands.

User Defined Functions

Under MPSF, execution of User Defined Functions takes place on the server side. The server executes as a batch job or started task and is not able to use TSO or ISPF services normally available to a client running under TSO/ISPF. You must therefore ensure that your User Defined Functions do not attempt to invoke TSO or ISPF services. In addition, you cannot access client specific information using MVS control blocks or services, as the information returned will relate to the server and not the client address space.

Appendix A

Example Batch Start

This is an example batch job that executes MPSF:

```
//MPSF0100   JOB  MPSF,'MPR BATCH SERVER',NOTIFY=SYSADM,REGION=0M,
//                CLASS=C,MSGCLASS=X,MSGLEVEL=(1,1),TIME=NOLIMIT
//****************************************************************
//*        STARTUP MPSF IN BATCH.
//****************************************************************
//MPSERVER EXEC PGM=MPR00,PARM='SERVER=NO'
//STEPLIB  DD   DISP=SHR,DSN=MP.MPR2500.LOADLIB
//MPIN     DD   DDNAME=SYSIN
//MPOUT    DD   SYSOUT=*
//MPRDIAG  DD   SYSOUT=*
//SYSUDUMP DD   SYSOUT=*
//STARTER  DD   DISP=SHR,DSN=MP.MPR2500.STARTER.MPAID
//MPAID    DD   DISP=SHR,DSN=MP.MPR2500.MPAID.LINEAR
//COM      DD   DISP=SHR,DSN=MP.MPR2500.COM.MPAID.LINEAR
//WRP3     DD   DISP=SHR,DSN=MP.MPR2500.WRP3.DICT.LINEAR
;
;    LOGON TO STARTER MPAID
;
LOGON STARTUP PASSWORD STARTUP MPAID-DDNAME STARTER;
;
;     DEFINE 100 SERVER TASKS FOR QPT, RPT AND CPT USE
;
SUBTASK DEFINE 100 SERVER;
;
;     START RPT FOR SHARED DIV SECONDARY MPAID
;
SUBTASK START  2     'LOGON COM PASSWORD COM SHARE MPAID-DDNAME COM;';
SUBTASK WAIT   2;
SUBTASK VIEW   2;
SUBTASK NAME   2     RPT-COM;
;
;     START RPT FOR SHARED DIV PRIMARY MPAID
;
SUBTASK START  3     'LOGON ADMIN PASSWORD ADMIN SHARE;';
SUBTASK WAIT   3;
SUBTASK START  3     'MP CONC COM;';
SUBTASK WAIT   3;
SUBTASK START  3     'CAPTURE START 50;';
SUBTASK VIEW   3;
SUBTASK NAME   3     RPT-PRIM;
;
```

```
;     START RPT FOR SHARED DIV REPOSITORY WRP3
;
SUBTASK START  4     'LOGON ADMIN PASSWORD ADMIN RPT;';
SUBTASK WAIT   4;
SUBTASK START  4     'MP CONC COM;';
SUBTASK WAIT   4;
SUBTASK START  4     'DICT WRP3 DIV SHARE;';
SUBTASK WAIT   4;
SUBTASK START  4     'AUTH WRP3;';
SUBTASK WAIT   4;
SUBTASK START  4     'CAPTURE START 30;';
SUBTASK VIEW   4;
SUBTASK NAME   4     RPT-WRP3;
;
;     START SERVER QPT
;
SUBTASK START  1     'LOGON ADMIN PA ADMIN QPT;';
SUBTASK WAIT   1;
SUBTASK START  1     'MP CONC COM;';
SUBTASK WAIT   1;
SUBTASK START  1     'SERVER START NAME MPSERV12 CTO 75 MDF;';
SUBTASK NAME   1     QPT-MPSF;
;
;     END OF SERVER STARTUP - NOW MAIN TASK WILL WAIT FOR SERVER
;     QUEUE PROCESSING TASK (QPT) TO END VIA 'SERVER END' COMMAND
;     ISSUED TO QPT FROM AN AUTHORISED (SYSAD) CLIENT TASK.
;
SUBTASK WAIT   1;                         WAIT FOR SERVER QPT TO END
SUBTASK VIEW   1;                         COLLECT SERVER LOG
SUBTASK START  1      LOGOFF;             END SERVER QPT
SUBTASK WAIT   1;                         WAIT FOR TASK TO END
;
;     SHUTDOWN WRP3 DICTIONARY RPT
;
SUBTASK START  4      'CAPTURE STOP';     STOP WRP3 REPOSITORY CAPTURE
SUBTASK WAIT   4;
SUBTASK VIEW   4;                         COLLECT RPT OUTPUT
SUBTASK START  4      LOGOFF;
SUBTASK WAIT   4;                         WAIT FOR TASK TO END
;
;     SHUTDOWN PRIMARY MPAID RPT
;
SUBTASK START  3      'CAPTURE STOP';     STOP PRIMARY MPAID CAPTURE
SUBTASK WAIT   3;
SUBTASK VIEW   3;                         COLLECT RPT OUTPUT
SUBTASK START  3      LOGOFF;
SUBTASK WAIT   3;                         WAIT FOR TASK TO END
;
;     SHUTDOWN SECONDARY MPAID RPT
;
SUBTASK VIEW   2;                         COLLECT RPT OUTPUT
SUBTASK START  2      LOGOFF;
SUBTASK WAIT   2;                         WAIT FOR TASK TO END
;
;    ALL TASKS HAVE ENDED, NOW TERMINATE MANAGER PRODUCTS RUN
;
LOGOFF;
//
```

Appendix B
Example Server Log

This is an example server log:

```
*****************************************************************************************
* SUBTASK-ID  : TASK-01    RETURN CODE: 4
* EXECUTED ON : 24 JUNE 1999 AT 08.51.32
*****************************************************************************************
DM14500I     MANAGER PRODUCTS SERVER MPSERV01 STARTED AT 08.51.32 ON 24 JUNE 1999
DM14540I     NEW CONCURRENT CONVERSATION LIMIT ACCEPTED
DM14544I     NEW GLOBAL TIMEOUT VALUE ACCEPTED
DM14511I     WEBOUT  (0001) CONNECTED AT 11.05.15 CLIENTS 001 MODE W
DM14511I     WEBOUT  (0002) CONNECTED AT 11.05.24 CLIENTS 002 MODE W
DM14511I     DEVMXC24(0003) CONNECTED AT 11.05.57 CLIENTS 003 MODE B
DM14511I     DEVMXC05(0004) CONNECTED AT 11.05.57 CLIENTS 004 MODE B
DM14511I     DEVMXC07(0005) CONNECTED AT 11.06.57 CLIENTS 005 MODE B
DM14511I     DEVMXC06(0006) CONNECTED AT 11.06.58 CLIENTS 006 MODE B
DM14512I     DEVMXC24(0003) ENDED     AT 11.07.27 CLIENTS 005 RETC 000 CPU 0010.75
DM14512I     DEVMXC05(0004) ENDED     AT 11.07.29 CLIENTS 004 RETC 000 CPU 0022.49
DM14511I     DEVMXC08(0007) CONNECTED AT 11.07.57 CLIENTS 005 MODE B
DM14511I     DEVMXC09(0008) CONNECTED AT 11.07.57 CLIENTS 006 MODE B
DM14512I     DEVMXC07(0005) ENDED     AT 11.08.51 CLIENTS 005 RETC 000 CPU 0008.14
DM14512I     DEVMXC06(0006) ENDED     AT 11.08.52 CLIENTS 004 RETC 000 CPU 0018.16
...
DM14511I     DEVMXC07(0015) CONNECTED AT 11.11.59 CLIENTS 009 MODE B
DM14511I     DEVMXC06(0016) CONNECTED AT 11.12.00 CLIENTS 010 MODE B
DM14511I     DEVMXC08(0017) CONNECTED AT 11.12.59 CLIENTS 011 MODE B
DM14514W     WEBOUT  (0002) TIMED-OUT AT 11.15.54 CLIENTS 010 RETC 000 CPU 0012.37
DM14512I     DEVMXC24(0013) ENDED     AT 11.17.01 CLIENTS 010 RETC 000 CPU 0016.59
DM14512I     DEVMXC05(0014) ENDED     AT 11.17.01 CLIENTS 009 RETC 000 CPU 0024.92
...
DM14511I     WEBOUT  (0033) CONNECTED AT 11.51.01 CLIENTS 001 MODE W
DM14517W     WEBOUT  (0033) CANCELLED AT 11.52.01 CLIENTS 001 RETC 000 CPU 0026.01
DM14511I     DEVMXC24(0034) CONNECTED AT 11.52.27 CLIENTS 001 MODE B
DM14512I     DEVMXC24(0034) ENDED     AT 11.52.56 CLIENTS 000 RETC 000 CPU 0014.41
DM14501I     MANAGER PRODUCTS SERVER MPSERV01 SUSPENDED AT 11.53.27 ON 24 JUNE 1999
DM14515W     DEVMXC96(0000) REJECTED  AT 11.53.49 CLIENTS 000 RETC 008 CPU 0028.13
DM14502I     MANAGER PRODUCTS SERVER MPSERV01 RESUMED AT 11.54.34 ON 24 JUNE 1999
DM14511I     DEVMXC24(0035) CONNECTED AT 11.55.00 CLIENTS 001 MODE B
DM14511I     DEVMXC07(0038) CONNECTED AT 11.56.01 CLIENTS 004 MODE B
...
DM14512I     DEVMXC11(0052) ENDED     AT 12.07.56 CLIENTS 003 RETC 000 CPU 0020.81
DM14512I     DEVMXC10(0051) ENDED     AT 12.07.56 CLIENTS 000 RETC 000 CPU 0011.25
DM14519I     MANAGER PRODUCTS SERVER MPSERV01 IN SHUTDOWN PHASE AT 12.08.57
DM14520I     MANAGER PRODUCTS SERVER MPSERV01 ENDED AT 12.08.58 ON 24 JUNE 1999
```

Index

ASG Worldwide Headquarters Naples Florida USA | asg.com